



Universidad
Carlos III de Madrid

Departamento de Informática

PROYECTO FIN DE CARRERA

INFERENCIA DE EMOCIONES A TRAVÉS DE DETECCIÓN CORPORAL Y FACIAL

Autor: Juan Antonio Rey Mendo

Tutor: Pedro Paredes Barragán

Leganés, octubre de 2012

Título: INFERENCIA DE EMOCIONES A TRAVÉS DE DETECCIÓN CORPORAL Y FACIAL

Autor: Juan Antonio Rey Mendo

Director: Pedro Paredes Barragán

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día __ de _____ de 20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos

A mi familia por comprenderme, darme su cariño y apoyarme, además de aguantarme en los períodos ajetreados y con apenas tiempo (que han sido más de uno y de dos) de estos seis años de carrera.

A mis viejos amigos Ismael, Pablo, David y Sara, con los que he compartido tanto que no cabría en la hoja y con los que he cambiado (esperemos que a mejor) a lo largo de muchos años y a otros no tan viejos como Elena, Jorge o Tachen.

A mis amigos y compañeros de la universidad por todos los viajes, las aventuras que nos quedan y el camino que hemos recorrido, especialmente a Javi (entre viejo amigo, compañero de clase y de equipo), José (que me dejó el *Kinect* y se mostró tan hospitalario como me tiene acostumbrado), Martín, Diego, Berlana, Aitor; además, a mis compañeros de prácticas que espero tengan un gran futuro porque profesionalidad les sobra: Carlos, Enrique y Santiago.

A mi equipo de fútbol, esperando que este año quedemos mínimo novenos de la liga de diez equipos.

A mi tutor de proyecto, Pedro Paredes, por ayudarme, orientarme y aclarar mis ideas a veces dispersas en este proyecto.

Y por último, mi más profundo agradecimiento a toda la comunidad docente de la Universidad Carlos III, especialmente a los profesores que nos han enseñado con ilusión y paciencia gran parte de lo que sabemos y han sabido renovarse en esta rama que está en un cambio constante, además de a los profesionales de la informática e investigadores que cada día hacen posible que ésta mejore.

Resumen

La emoción, del latín *emotio* (impulso) es la reacción a un estímulo externo a partir de una experiencia previa, además de la base de la comunicación no verbal.

Se está produciendo un cambio en los paradigmas de interacción entre las personas y los ordenadores (de su estudio se ocupa la rama conocida como HCI, por sus siglas en inglés): actualmente las interacciones realizadas son más naturales (dispositivos táctiles) y normalmente a una distancia mayor (control a distancia de dispositivos).

Existe una problemática relacionada con el cambio mencionado anteriormente: para el reconocimiento de las emociones, se ha usado clásicamente el reconocimiento facial; la interacción actual con los dispositivos a una distancia mayor demanda la exploración de otras alternativas para dicho reconocimiento, mediante la voz o los gestos realizados por un usuario. Esta última alternativa será la estudiada a lo largo de este proyecto.

Las aplicaciones de dicho estudio serían diversas: por ejemplo en el área de experiencia de usuario, en cuanto a que supondría una nueva manera de evaluar la usabilidad y de satisfacer plenamente sus necesidades, en combinación con otras técnicas de extracción del conocimiento como entrevistas o encuestas. Otro ejemplo importante sería en labores relacionadas con el área de recursos humanos que requieran un examen minucioso de los estados de ánimo de un individuo, como podría ser una entrevista de trabajo.

El objetivo de este proyecto es concentrarse en la **detección corporal y facial para la inferencia de emociones** mediante la visión por computador.

Para la tarea descrita anteriormente, se empleará el dispositivo *Kinect* de Microsoft, que ofrece un kit de desarrollo software (SDK) para dicho dispositivo que facilita la labor de reconocer tanto el esqueleto humano como las expresiones faciales y, por tanto, el posterior análisis de ellos.

Palabras clave: computación afectiva, detección corporal, detección facial, inferencia de emociones, Kinect, visión por computador.

Abstract

Emotion, a word derived from Latin “*emotio*” (impulse) is the reaction to an external stimulus, generated from a previous experience, as well as the basis of non-verbal communication.

The interaction paradigms between humans and computers are changing (the branch involved in this study is known by his acronym, HCI): nowadays the accomplished interactions are more natural (e.g. tactile devices) and normally at a greater distance (e.g. remote control devices).

There is a problematic related to change previously mentioned: to emotion recognition, there is classically used the facial tracking, but the prevailing interaction with devices at a greater distance demands the exploration of other alternatives to accomplish this recognition, through voice or gestures done by an user. The last alternative will be studied along this project.

The applications of this study are several: for example, in User Experience area it could suppose a new way to evaluate the usability and satisfy completely the user needs, in combination with other knowledge-extraction techniques such as interviews or surveys. Another important example could be in tasks related with Human Resources area that require a deep exam of the emotional states of an individual, such as a job interview.

Since Affective Computing is such a broad field, the objective of this project is to focus on the **face and body tracking to inferring emotions**, by means of computer vision.

For the analysis mentioned before, Microsoft *Kinect* device will be used with his SDK (Source Development Kit) that makes easier the task of recognize both the human skeleton as well as the facial expressions and, hence, the later analysis of it.

Keywords: affective computing, body tracking, computer vision, emotion inference, face tracking, Kinect.

Índice general

1.	Introducción	9
1.1.	Planteamiento del problema	9
1.2.	Objetivos	11
1.3.	Estructura de la memoria.....	11
1.4.	Pasos realizados en el proyecto	12
2.	Estado del arte.....	14
2.1.	HCI y Computación Afectiva.....	14
2.2.	Breve introducción a las emociones básicas.....	16
2.3.	Experimentos previos en la computación afectiva	17
2.4.	Software para reconocimiento de emociones	18
2.4.1.	ACUME	18
2.4.2.	AutoTutor	19
2.4.3.	FaceSense	21
2.5.	Mecanismos y dispositivos utilizados para la extracción de emociones	21
2.5.1.	Actividad Electrodermal (EDA)	21
2.5.2.	Voz.....	23
2.5.3.	Imagen.....	25
2.6.	Tecnología hardware: <i>Kinect para Xbox 360</i> y <i>Kinect para Windows</i>	26
2.7.	Especificaciones de <i>Kinect para Xbox 360</i>	28
2.8.	Entornos de desarrollo para el trabajo con <i>Kinect</i>	29
2.8.1.	<i>OpenNI</i>	29
2.8.2.	<i>Kinect para Windows SDK</i>	30
3.	Análisis del sistema.....	32
3.1.	Definición del problema	32
3.2.	Definición del sistema	34
3.2.1.	Alcance del sistema	34
3.2.2.	Identificación del entorno tecnológico	35
3.2.3.	Identificación de usuarios participantes y finales	35
3.3.	Clasificación de emociones según los gestos del usuario	36
3.4.	Establecimiento de requisitos software.....	38
3.4.1.	Requisitos funcionales.....	39
3.4.2.	Requisitos no funcionales	43

3.5.	Casos de uso.....	46
3.6.	Definición de interfaces de usuario. Alternativa escogida.....	48
4.	Diseño del sistema.....	51
4.1.	Diagrama de paquetes	51
4.2.	Análisis de clases	53
4.2.1.	Primera parte del diagrama de clases.....	55
4.2.2.	Segunda parte del diagrama de clases.....	60
4.3.	Algoritmos usados.....	63
4.3.1.	Algoritmo de detección de poses.....	63
4.3.2.	Algoritmo de detección de movimiento	64
4.3.3.	Algoritmo de detección facial	66
4.3.4.	Algoritmo de inferencia de emociones	66
4.4.	Interfaz de usuario utilizada.....	67
5.	Resultados, trabajo futuro y conclusiones	69
5.1.	Resultados	69
5.2.	Trabajo futuro	73
5.3.	Conclusiones.....	73
	Bibliografía	75
	Anexo A: Glosario de términos.	78
	Anexo B: Planificación	80
	Anexo C: Presupuesto	82
	Gastos de personal	82
	Gastos de hardware.....	82
	Gastos de software y licencias.....	83
	Gastos de material fungible.....	84
	Resumen de gastos totales	84
	Plantilla de resumen	85

Índice de Ilustraciones

Ilustración 1: Enfoque tradicional del análisis emocional mediante visión por ordenador	10
Ilustración 2: Enfoque del problema conforme a lo diseñado en el proyecto.....	10
Ilustración 3: Emociones básicas según Paul Ekman (1972)	16
Ilustración 4: Emociones básicas observadas para el PFC	17
Ilustración 5: Captura de pantalla de ACUME.....	19
Ilustración 6: Sistema BPMS (medidor de presión corporal)	20
Ilustración 7: Explicación de funcionamiento de AutoTutor.....	20
Ilustración 8: Q sensor: sensor afectivo	22
Ilustración 9: Sistema FEEL al completo.....	22
Ilustración 10: Distribuciones normales de la frecuencia fundamental de la voz	23
Ilustración 11: Relación de parámetros de la voz con las emociones principales	24
Ilustración 12: Medición del pulso cardíaco mediante cámara	26
Ilustración 13: Diferencias entre el modo por defecto y el modo cercano de Kinect	27
Ilustración 14: Kinect para Xbox 360.....	28
Ilustración 15: Vista por capas del entorno OpenNI	29
Ilustración 16: Vista por capas del entorno Kinect for Windows SDK	31
Ilustración 17: Esquema del problema a resolver.....	33
Ilustración 18: Diagrama de Casos de Uso	46
Ilustración 19: Primera alternativa de diseño de la aplicación	48
Ilustración 20: Segunda alternativa de diseño de la aplicación	49
Ilustración 21: Tercera alternativa de diseño de la aplicación.....	50
Ilustración 22: Diagrama de paquetes del sistema	51
Ilustración 23: Diagrama de clases simplificado	53
Ilustración 24: Diagrama detallado de clases del sistema	54
Ilustración 25: Articulaciones del esqueleto reconocidas por <i>Kinect</i>	63
Ilustración 26: Algoritmo de detección de pose corporal.....	64
Ilustración 27: Algoritmo de detección de movimiento corporal.....	65
Ilustración 28: Ejemplo de unidad de animación	66
Ilustración 29: Interfaz de usuario final	67
Ilustración 30: Ejemplo del XML generado por la aplicación	70
Ilustración 31: Detección de aversión	71
Ilustración 32: Detección de sorpresa.....	71
Ilustración 33: Detección de alegría.....	72
Ilustración 34: Detección de tristeza.....	72
Ilustración 35: Diagrama de Gantt de la planificación estimada del proyecto	81
Ilustración 36: Diagrama de Gantt con la duración real del proyecto	81

Índice de Tablas

Tabla 1: Definición de partes corporales implicadas y posibles estados	37
Tabla 2: Clasificación final de las emociones respecto a los gestos detectados	37
Tabla 3: RS-01 Mostrar imagen del usuario	39
Tabla 4: RS-02 Mostrar el porcentaje de cada emoción	39
Tabla 5: RS-03 Mostrar el reconocimiento corporal y facial	40
Tabla 6: RS-04 Iniciar interacción con el usuario	40
Tabla 7: RS-01 Generar XML con movimientos del usuario	41
Tabla 8: RS-06 Finalizar la generación del XML con movimientos del usuario	41
Tabla 9: RS-07 Mostrar fiabilidad de medición	42
Tabla 10: RS-08 Mostrar emoción global del usuario	42
Tabla 11: RS-09 Tiempo de respuesta del sistema	43
Tabla 12: RS-10 Número de usuarios simultáneos	43
Tabla 13: RS-11 Especificación de algoritmos de la aplicación	44
Tabla 14: RS-12 Interfaz intuitiva	44
Tabla 15: RS-13 Resoluciones de la aplicación	45
Tabla 16: RS-14 Lenguaje de la aplicación	45
Tabla 17: RS-15 Librerías usadas para <i>Kinect</i>	45
Tabla 18: CU-01 Interactuar con sistema	47
Tabla 19: CU-02 Evaluar gestos observados	47
Tabla 20: CMD-01 Clase GestosAreaCorporal	56
Tabla 21: CMD-02 Clase GestosCabeza	57
Tabla 22: CMD-03 Clase GestosEspalda	57
Tabla 23: CMD-04 Clase GestosHombros	58
Tabla 24: CMD-05 Clase GestosBrazos	58
Tabla 25: CMD-06 Clase GestosBoca	59
Tabla 26: CMD-07 Clase GestosCejas	59
Tabla 27: CMD-08 Clase MainWindow	60
Tabla 28: CMD-09 Clase MotorInferencias	61
Tabla 29: CMD-10 Clase Utilities	62
Tabla 30: Algoritmo de clasificación de gestos en emociones	67
Tabla 31: Planificación estimada del proyecto	80
Tabla 32: Duración real del proyecto	81
Tabla 33: Gastos de personal	82
Tabla 34: Gastos de hardware	83
Tabla 35: Gastos de software	83
Tabla 36: Gastos totales asociados al proyecto	84

1. Introducción

En el desarrollo de las habilidades sociales las emociones juegan un papel muy importante, ya que nos capacitan a entender cómo se siente la otra persona. Esto ha sido un objeto de estudio tradicional de la psicología; más tarde, este objeto de estudio se vio reflejado en una rama de la informática derivada fundamentalmente de la Interacción Persona Ordenador (de ahora en adelante HCI por sus siglas en inglés) llamada “Affective Computing”, como será expuesto en el capítulo 2.

Han sido numerosos los trabajos en los que se han intentado analizar las acciones de los usuarios para así poder inferir sus emociones a través de diversos medios; fundamentalmente por medio del análisis de la voz o de la visión por ordenador (en cuyo caso nos vamos a centrar).

Dentro de la visión por ordenador, ha existido un intento por inferir las emociones a través de un proceso previo y necesario: el análisis de los movimientos del cuerpo y de la cara (mediante los que se manifiesta la exteriorización de las emociones), que será presentado en el capítulo 3 para dar una noción intuitiva sobre lo que se ha llevado a cabo.

1.1. Planteamiento del problema

La gran mayoría de los programas que intentaban reconocer las emociones mediante visión por ordenador se basan exclusivamente en el análisis facial del usuario, lo que significa que, a menos que se tenga una cámara con una resolución muy alta, es un requisito indispensable que el usuario esté muy cerca de la misma para poder analizar correctamente sus emociones.

Las nuevas formas de interacción con los dispositivos, más naturales y a una distancia mayor, hacen que dichos programas basados exclusivamente en análisis facial empiecen a no ser los más adecuados para el reconocimiento de emociones.

Por medio de un estudio sobre cómo se diseñan los avatares para reproducir emociones (tristeza, alegría, enfado) se han llevado a cabo ciertas deducciones para permitir cambiar un enfoque del análisis emocional en principio algo cerrado, mediante el análisis de las emociones no sólo basándose en los gestos faciales, sino **también en los gestos corporales**.

De un vistazo rápido, en la ilustración 1 se muestra el enfoque tradicional del problema mediante técnicas de visión artificial, y en la ilustración 2 se muestra cómo se va a abordar el problema en esta ocasión (mediante gestos y a una distancia mayor de lo que es habitual).

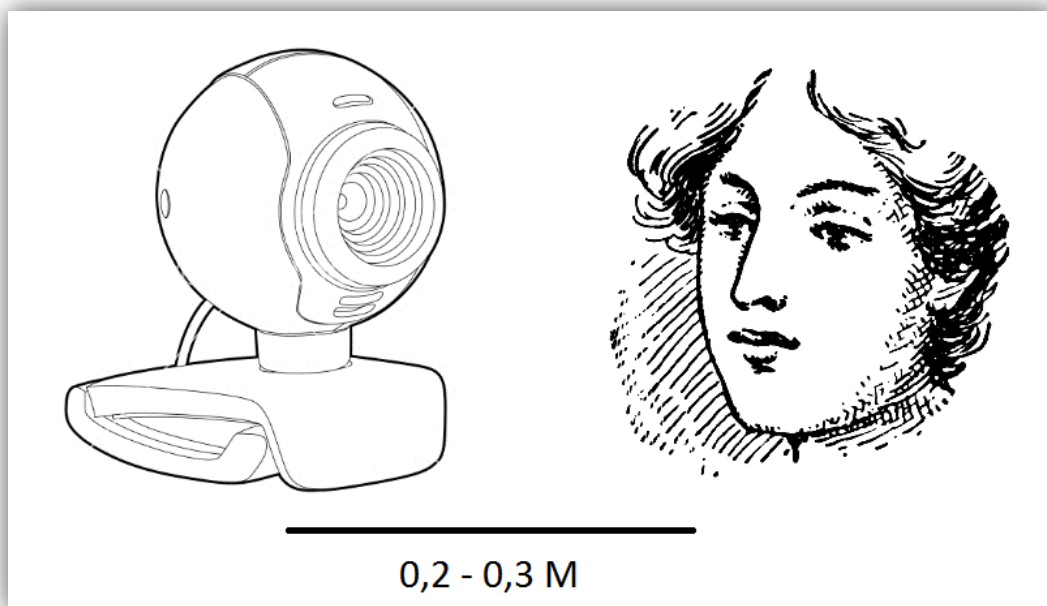


Ilustración 1: Enfoque tradicional del análisis emocional mediante visión por ordenador

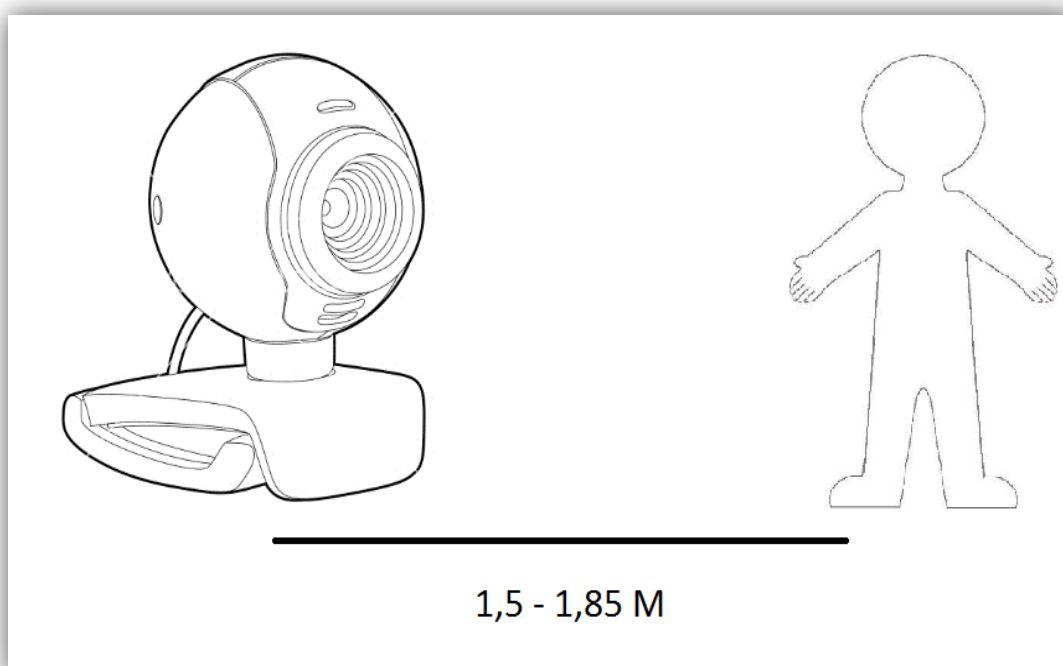


Ilustración 2: Enfoque del problema conforme a lo diseñado en el proyecto

1.2. Objetivos

El objetivo principal del proyecto sería conseguir una **inferencia de emociones a partir de patrones definidos y de manera contrastable (almacenando dicha información)**; debido a que las emociones entran en la relatividad de cada persona, este objetivo principal tendrá un matiz subjetivo en su efectividad detectando las emociones.

Los objetivos secundarios para conseguir el proyecto serían:

- Conseguir hacer detección precisa del cuerpo:
 - Analizar las poses del usuario (esto es, análisis estático de las articulaciones observadas en el esqueleto).
 - Analizar los movimientos del usuario (análisis dinámico comparando varias imágenes de las articulaciones y su evolución para inferir un movimiento).
- Conseguir hacer detección facial:
 - Analizar la información proporcionada por el entorno de desarrollo de *Kinect*.
 - Detectar los movimientos de las cejas y la boca a partir de dicha información.
- Almacenar la información extraída en un soporte secundario:
 - Almacenar las listas de movimientos de esqueleto del usuario y cara.
 - Unirlas para generar un XML que expresa el estado preciso de los diferentes movimientos ocurridos a cada segundo.
- Inferir emociones a partir de la detección gestual.
 - Asignar pesos de una manera eficaz a cada uno de los estados descritos para poder inferir una emoción.
 - Expresar dichas emociones en formas de porcentaje en cada iteración para poder ver con claridad qué ocurre en el sistema.
 - Resumir el estado actual del usuario almacenando una emoción global.

1.3. Estructura de la memoria.

Esta memoria contará con cinco capítulos, bibliografía y un anexo. A continuación se describe brevemente cada uno de ellos:

- Capítulo 1: Introducción: planteamiento del problema actual, objetivos primarios y secundarios y breve descripción de los pasos llevados a cabo para el mismo.
- Capítulo 2: Estado del arte: breve contextualización del problema en el área que lo estudia, nociones básicas sobre las emociones, herramientas usadas actualmente en problemas similares o parecidos y exploración de la tecnología *Kinect*.

- Capítulo 3: Análisis del sistema: definición concreta del problema, opciones posibles para su diseño e implementación, extracción de requisitos generales y análisis entre las posibles alternativas de interfaz.
- Capítulo 4: Diseño del sistema: diseño detallado del sistema: diagramas de paquetes y clases, análisis de clases, algoritmos e interfaz de usuario usados.
- Capítulo 5: Resultados, líneas futuras y conclusiones: descripción sucinta sobre los resultados obtenidos y las conclusiones; en las líneas futuras, figurarán las posibles ampliaciones que se puedan realizar respecto a lo expuesto en los capítulos anteriores.
- Bibliografía: contiene las fuentes consultadas y referenciadas a lo largo del proyecto.
- Anexos: aquí se incluyen: el glosario, el presupuesto y la planificación realizada.

1.4. Pasos realizados en el proyecto

Aquí se expone a modo de resumen la metodología seguida para realizar el proyecto, que ha constado de los siguientes pasos:

- 1) Estudio sobre la parte no técnica del problema.
 - Obtención de información y lectura sobre las emociones.
 - Revisión de distintos trabajos relacionados con las emociones.
 - Observación de carencias sobre el lenguaje no-verbal del cuerpo desde una perspectiva informática.
- 2) Estudio sobre la parte técnica del problema.
 - Examinar posibles alternativas de afrontar el problema (en un marco tecnológico).
 - Selección de una tecnología concreta (en este caso, *Kinect* debido a su relación calidad / precio y a la disposición de un entorno concreto de desarrollo, así como de la colaboración por parte de Microsoft).
 - Diseño genérico de clases básicas que debería contener el problema.
- 3) Comienzo de documentación.
 - Diseño y clasificación más exhaustivas.
 - División de nuevas necesidades.
 - Salida de un entorno de desarrollo más avanzado que permite la realización de un análisis facial con cierta precisión.
 - Estado del arte, análisis y desarrollo contemplados.

- 4) Implementación de la primera versión.
 - Diseño exclusivamente basado en los gestos corporales.
 - División y clasificación de la información según **el tipo de emoción** extraída.
 - Comienzo de exploración de diversas alternativas para afrontar el problema, seleccionándose finalmente el uso del entorno de desarrollo proporcionado por Microsoft.
 - Punto de partida: seis emociones básicas.

- 5) Implementación de la segunda versión.
 - Diseño basado en los gestos corporales y faciales.
 - Clasificación de la información según **las partes del cuerpo**.
 - Reducción de las seis emociones básicas a cuatro, para una mejor clasificación.
 - Ponderación de los gestos actuales para la inferencia de una de las cuatro emociones.

- 6) Final de documentación (conclusiones, bibliografía y anexos), revisión y pruebas de la implementación.

- 7) Fases de pruebas finales de implementación y revisión de la documentación.

2. Estado del arte

El estado del arte comprende los últimos avances en la situación actual del campo a investigar, además de las tecnologías que se utilizarán en el proyecto.

Existe una clara división en el estudio del mismo, que abarca fundamentalmente los siguientes puntos:

- Interacción persona-ordenador (*HCI*) y computación afectiva.
- Breve introducción a las emociones básicas.
- Experimentos previos en la computación afectiva.
- Software para reconocimiento de emociones.
- Mecanismos y dispositivos utilizados para la extracción de emociones.
- Tecnología hardware: *Kinect* para Xbox 360 y *Kinect* para Windows
- Especificaciones de *Kinect*.
- Entornos de desarrollo para el trabajo con *Kinect*.

2.1. HCI y Computación Afectiva.

En primer lugar, se ofrecerá una perspectiva en la que se contextualizará el problema para más tarde analizar la relación entre la interacción persona-ordenador y la computación afectiva.

La HCI o Interacción persona-ordenador, trata sobre el intercambio de información producido entre ambos. En este sentido, tiene muchas ocupaciones alrededor. Una de las misiones más claras de esta disciplina es la de crear aplicaciones cada vez más usables (característica referida a la facilidad de uso y aprendizaje de un sistema, así como la satisfacción del usuario).

El uso de dispositivos que facilitan la interacción con el usuario, como pueda ser el reconocimiento de gestos con la webcam, potencia mucho la usabilidad de un sistema. Incluso muchas veces, en lugar de atajos de teclado, podrían crearse atajos gestuales que sustituyeran estas necesidades para aumentar el ritmo de trabajo, pero es especialmente en la **satisfacción del usuario** donde encontramos una relación directa con la computación afectiva.

Según Rosalind W. Picard, fundadora del grupo de investigación sobre Computación Afectiva, y autora del libro con el mismo título [2], la computación afectiva es una disciplina encargada de evaluar la relación entre el ordenador y los hechos que producen, generan o influyen sobre las emociones.

Además, Picard es la encargada de establecer la relación entre ambas disciplinas en su *paper* (texto científico) “Affective Computing for HCI” [3]; la relación sería la siguiente:

- Reducción de la frustración del usuario.
- Permitir una comunicación cómoda de las emociones del usuario.
- Desarrollar infraestructura y aplicaciones para manejar información afectiva.
- Construir herramientas para desarrollar habilidades sociales.

Como podemos ver, estos objetivos tienen una estrecha relación unos con otros; este proyecto se centrará fundamentalmente en el manejo de la información afectiva expresada por el usuario, a través de la extracción de datos sobre sus gestos actuales.

Del tercer objetivo mencionado (el manejo de información afectiva) se podrían desarrollar los otros tres objetivos, del modo que a continuación se expone:

- Reducción de la frustración del usuario: cuando se observe al usuario efectuando pruebas con un prototipo del sistema, en el área de experiencia de usuario, será muy importante tener bajo control que no existan emociones negativas que provocarán más tarde dicha frustración; con este proyecto, esto podría evitarse.
- Permitir una comunicación cómoda de las emociones del usuario: el sistema desarrollado debería ser capaz de inferir las emociones sin que el usuario deba esforzarse excesivamente o forzar las emociones. En este sistema esto no es algo trivial, ya que la distancia del usuario (bastante mayor que en otros proyectos, como se expuso en la comparativa de la Ilustración 1 con la Ilustración 2) dificulta esta tarea.
- Construcción de herramientas para desarrollar habilidades sociales: a partir de la observación de las emociones actuales conforme a algunos estímulos, podrían crearse aplicaciones que sirvieran de correctivos a una experiencia que nos suscite una determinada emoción; esto es lo que en psicología se conoce por “reprogramación mental” y ayuda a las personas a poder desenvolverse en entornos más heterogéneos; para más información, la rama psicológica encargada de estudiar esto se denomina programación neurolingüística (PNL) y aquí hay una breve explicación de lo expuesto anteriormente [4].

2.2. Breve introducción a las emociones básicas.

No existe una aproximación exacta en cuanto al número de emociones básicas; las fuentes de información consultadas hablan de entre 4 y 11 emociones básicas, así que se expone una clasificación suficientemente compacta que daría lugar a las emociones más básicas.

En particular, la clasificación básica que realiza el psicólogo Paul Ekman, uno de los 100 más reputados del siglo XX [5] en el año 1972 es la siguiente:

- Miedo.
- Tristeza.
- Alegría.
- Cólera.
- Disgusto.
- Sorpresa.

En 1990 esta clasificación se vio modificada y extendida, pero se considera suficiente la clasificación básica expuesta arriba; ambas son las más usadas en el reconocimiento de las emociones por parte de una máquina [6].

En la Ilustración 3 se encuentra el conjunto completo de las emociones básicas; este ha sido reducido de cara al proyecto, donde sólo se reconocerán las emociones expuestas en la Ilustración 4:

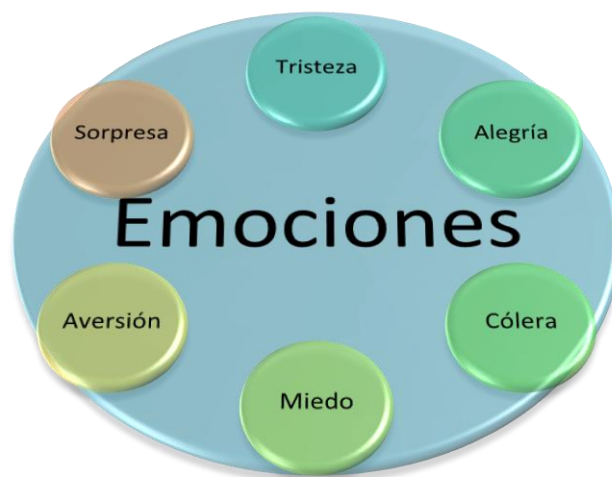


Ilustración 3: Emociones básicas según Paul Ekman (1972)



Ilustración 4: Emociones básicas observadas para el PFC

2.3. Experimentos previos en la computación afectiva

Se han realizado una extensa cantidad de estudios al respecto [7] [8] [9] pero la mayoría de ellos están relacionados con el reconocimiento de expresiones faciales, hablando de la captación de varias emociones básicas y a su vez emociones secundarias ligadas a las mismas (dependiendo de la complejidad a la que lleguen).

Debe tenerse en cuenta que no fue hasta 1992 cuando un estudio de Nalini Ambady y Robert Rosenthal, de la universidad de Harvard determinó que la combinación de los gestos faciales y los gestos del cuerpo nos daban una precisión para predecir el estado de ánimo de un 54% [10]; a pesar de ello, la literatura relacionada no se centra en la información expresiva del cuerpo hasta 2003, con el artículo “Expressive Interfaces” (A. Camurri et al.), donde presenta modelos computacionales y algoritmos relacionados con el reconocimiento del cuerpo [11].

El análisis expuesto en [8] nos muestra que, aparte de lo mencionado previamente sobre la computación afectiva como conjunto de gestos, algunas de las características de estos más allá del movimiento pueden ser **indicadores** sobre las emociones; en concreto, que en función de la rapidez con la que se mueva el brazo, puede denotar que los movimientos rápidos del brazo están asociados con una situación de enfado o alegría, mientras que los movimientos lentos se asocian con una situación de tristeza (no ha sido tenido en cuenta realizando este proyecto).

Asimismo, se ha realizado [12] una clasificación de la información afectiva que se percibe en tres categorías: actuada (se le pide a una persona que muestre las emociones para comprobar que el sistema funciona), inducida (por ejemplo, se pone a la persona del experimento a ver una película y analizar sus emociones) y espontánea (un terreno de la vida real en el que la persona muestra sus emociones); este proyecto se centrará especialmente en la información afectiva **actuada**, no descartando las anteriores en desarrollos futuros.

Para el reconocimiento de gestos del cuerpo también existe una subdivisión en tres categorías [13]: basada en un modelo tridimensional (observando el movimiento previo diseño de un modelo tridimensional que exponga el cuerpo y sus posibles expresiones, usado por ejemplo en la industria de los videojuegos), basada en la apariencia (modelo bidimensional, basado en la escala de grises) y basada en el movimiento (usando simplemente la información del movimiento sin información del cuerpo físico). El modelo que recrea *Kinect* es un modelo tridimensional si se usa el sensor de profundidad (como veremos más adelante), pero en el reconocimiento de gestos lo más apropiado es un análisis **basado en el movimiento con información del cuerpo físico** (que el dispositivo *Kinect* nos permite con su identificación del esqueleto).

El análisis propuesto hasta ahora se refería al reconocimiento de expresiones **corporales** como factor diferencial; sin embargo, se ha considerado muy relevante realizar un análisis de la boca y las cejas que proporcionan información adicional, ya que será determinante a la hora de evaluar las emociones.

2.4. Software para reconocimiento de emociones

Se han realizado numerosos experimentos en esta área desde la fecha de su creación (1997) [14]; aquí se procede a exponer los que guardan una mayor relación con el proyecto actual:

2.4.1. ACUME

Esta herramienta se trata de un potente analizador facial para conseguir extraer las emociones; su principal ventaja es la facilidad de visualización y su potencia y precisión en el análisis de las distintas **unidades de animación**. Una unidad de animación (conocida como AU en inglés) analiza las distintas variaciones que se producen en la cara en comparación con su forma neutral. La descripción completa del programa se encuentra en [15].

En este programa en concreto, aparecen un total de 27 unidades de animación analizadas, que son exactamente las mismas que aparecen en el modelo facial Candide3 [16]; algo mejor

que el Face Tracking que posee Microsoft, que sólo cuenta por el momento con 6 unidades de animación [17], y que han sido utilizadas en el proyecto.

Sería muy positivo su uso en la detección de los llamados “microgestos”; es decir, variaciones faciales muy pequeñas que permiten aumentar mucho la precisión a la hora de describir las emociones; sin embargo, para la detección de emociones básicas, en el proyecto se demuestra que las 6 unidades de animación que incorporó Microsoft son suficientes.

En la Ilustración 5 figura una captura del programa en cuestión:

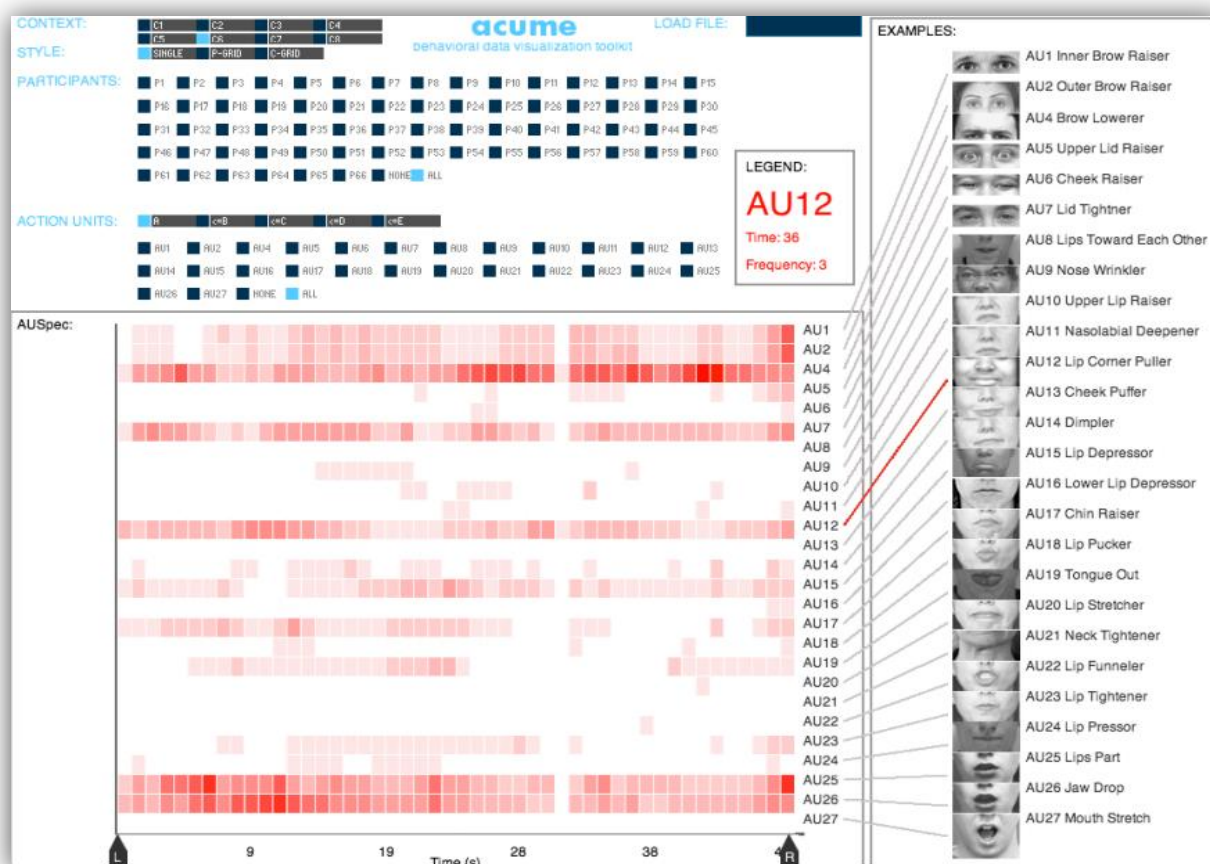


Ilustración 5: Captura de pantalla de ACUME

2.4.2. AutoTutor

Se trata de un software que detecta y responde a los estados de ánimo del alumno. Usa un análisis de la postura del alumno sentado, examina sus ojos y analiza la modulación de su voz para la inferencia, por lo que sería de gran utilidad en un momento (no muy lejano) en el que muchas de las clases se impartirán presumiblemente en la red. La documentación relativa puede verse en [18].

Además, en cuanto a la respuesta a los estados, es capaz de modelar muchas respuestas distintas en función de cada parámetro suministrado. De los dispositivos usados para la detección el más llamativo sería el que detecta la postura de la silla; esto consiste en un mecanismo compuesto por una matriz de sensores que miden la presión ejercida en cada uno de ellos; este mecanismo se llama BPMS (Sistema de Medida de la Presión Corporal, traduciendo sus siglas al español) y es como figura en la Ilustración 6:

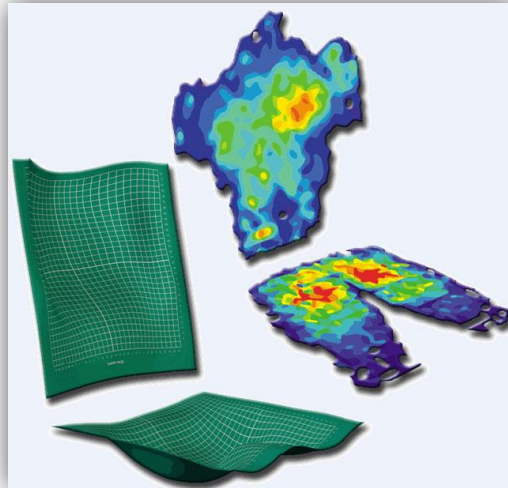


Ilustración 6: Sistema BPMS (medidor de presión corporal)

Un ejemplo de este programa figura en la Ilustración 7:

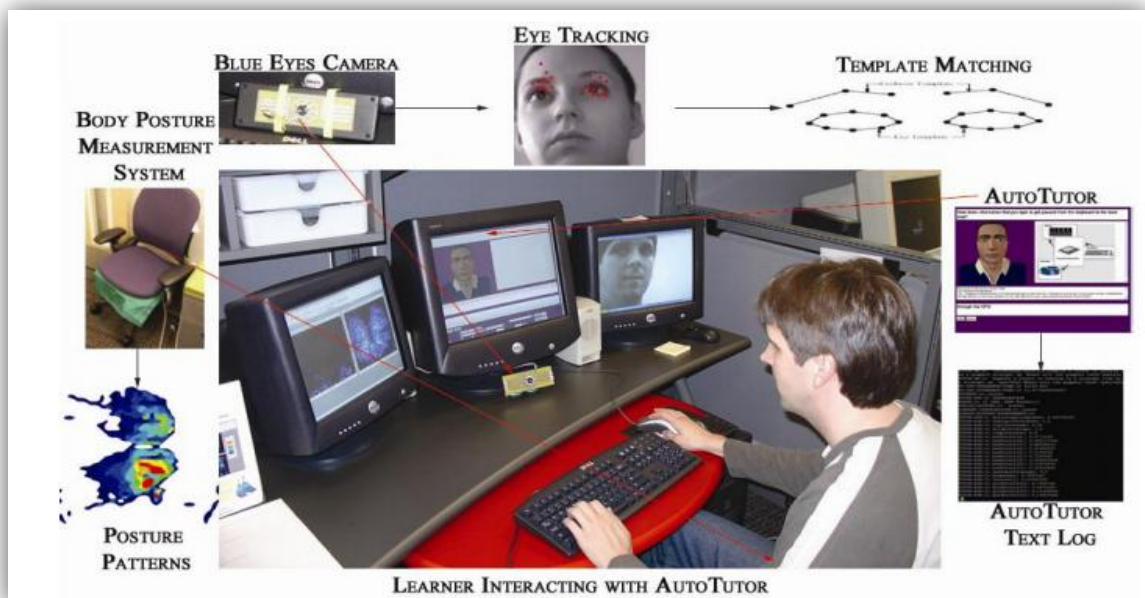


Ilustración 7: Explicación de funcionamiento de AutoTutor

2.4.3. FaceSense

Aplicación que no sólo se basa en las Unidades de Animación descritas anteriormente y en los gestos, sino que además tiene un añadido, la observación de **estados mentales**.

En las unidades de animación se usa la geometría; en los gestos, realmente sólo analiza los gestos de la cara mediante lo que se denominan Modelos Ocultos de Markov (HMM), que consiste en un modelo estadístico en el que intenta determinarse el valor de algunos parámetros desconocidos a partir de parámetros observables. Por último, para detectar los estados mentales, lo realiza mediante una Red Bayesiana Dinámica (DBN), que consiste en que las típicas inferencias de una red bayesiana se propagan en el tiempo y se crean diferencias entre dependencias en el tiempo; es así como construye este modelo, analizando en el tiempo la variación de los gestos de la cara y haciendo una especie de inferencia global; en el presente proyecto, se usará un mecanismo bastante menos técnico: la media de las emociones medidas en cada imagen grabada, que funciona suficientemente bien.

Cada uno de estos estados mentales proviene de un entrenamiento previo, y se pueden definir nuevos estados entrenando otros conjuntos.

La descripción parcial de este programa se encuentra en [19].

2.5. Mecanismos y dispositivos utilizados para la extracción de emociones

En este apartado se expondrán los diferentes mecanismos y dispositivos que se utilizan actualmente en el proceso de la inferencia de las emociones; más tarde se centrará la explicación en el dispositivo concreto utilizado a partir del [apartado 2.6](#).

2.5.1. Actividad Electrodermal (EDA)

Este tipo de dispositivos toman medidas sobre el nivel de activación del sujeto, fijándose en su sistema nervioso simpático (es decir, el que nos prepara para la acción) [20]. Este sistema se asocia con un estímulo emocional no neutro, por lo que tiene utilidad en la inferencia de algunos estados emocionales.

Explicado de una manera sucinta, una situación que tiene ligada una alta carga emocional (normalmente una situación asociada con el estrés o la tensión) hará que nuestras glándulas sudoríparas segreguen sales, con lo que aumentará la conductividad de nuestra piel.

En caso opuesto, cuando el sujeto esté en calma, dicha conductividad disminuirá, lo que será muy útil como medida de posibles emociones negativas.

Un ejemplo muy representativo del uso de estos sistemas se encuentra en [21]; se trata de FEEL, y se compone de:

- Un sensor de estrés que funciona con esta medida del EDA, un detector de temperatura de la piel y un acelerómetro para sus inferencias (como figura en la Ilustración 8):



Ilustración 8: Q sensor: sensor afectivo

- Un sistema que monitoriza eventos a partir de la información extraída de un móvil (mide mediante el sensor la respuesta a los estímulos de éste). Un ejemplo del sistema completo sería la Ilustración 9:

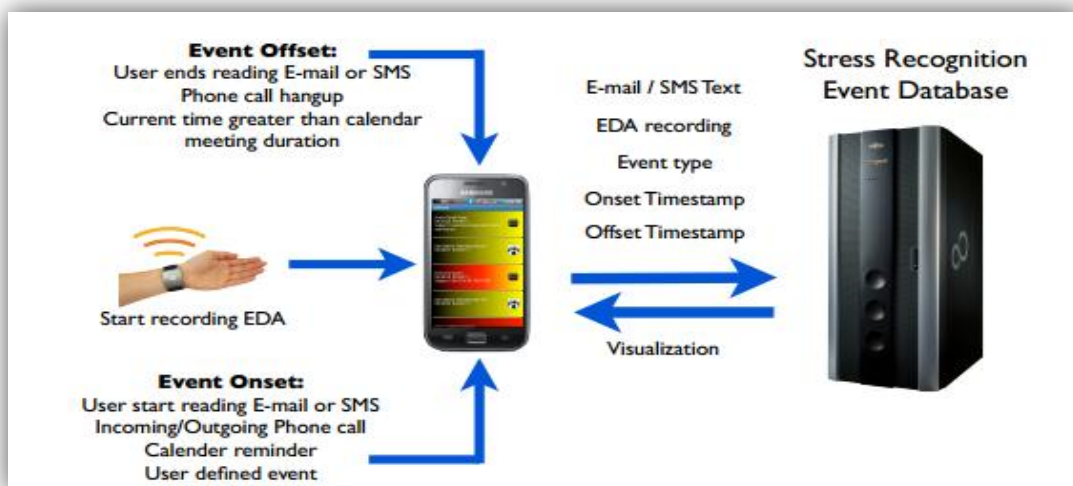


Ilustración 9: Sistema FEEL al completo

2.5.2. Voz

En estos mecanismos se tienen en cuenta las características de la voz; normalmente extrayéndose las emociones por medio de comparaciones con grabaciones previamente hechas.

Para el análisis de la voz se usan normalmente los siguientes parámetros [22]:

- Frecuencia fundamental (“pitch”): frecuencia a la que oscilan nuestras cuerdas vocales al emitir sonido en un estado no alterado. Su valor medio expresa el nivel de excitación del locutor (a mayor frecuencia, mayor excitación), y sus fluctuaciones (lo que normalmente se conoce por variaciones del tono de voz) son más abruptas en emociones negativas y más suaves si dichas emociones son positivas. Esta frecuencia aparece en la Ilustración 10:

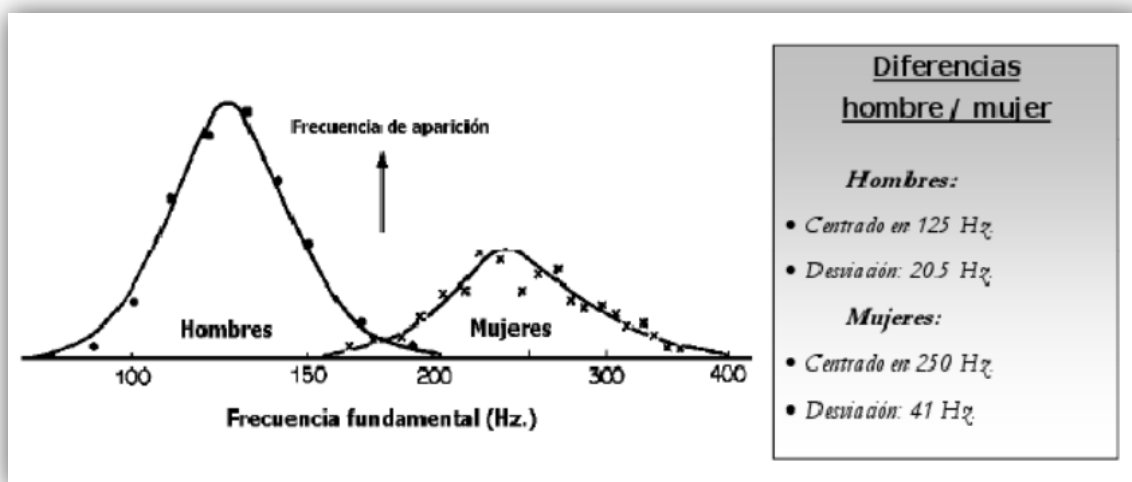


Ilustración 10: Distribuciones normales de la frecuencia fundamental de la voz

- Duración: se refiere al ritmo y velocidad del habla. La velocidad crecerá conforme al estado de excitación del locutor, y en el ritmo se puede establecer una asociación entre unas pausas más largas y un estado que tienda a la depresión.
- Calidad de voz: los factores más influyentes son las irregularidades vocales y la energía (intensidad) de la voz. Sobre las irregularidades vocales, se refieren a un cambio brusco (se establece el ejemplo de un paso al susurro relacionado con emociones como la pena), y en cuanto a la intensidad de la voz, una cantidad alta se relaciona con agitación y una baja con depresión y pena.

Para tomar las medidas relacionadas con la voz, normalmente se hace uso de aparatos matemáticos de complejidad alta, por lo que no es el objetivo de este proyecto su descripción

minuciosa en este apartado; sí se dejará una tabla ilustrativa [23] como referencia para entender cómo suelen estar graduados los sistemas que miden las emociones por voz (Ilustración 11):

	<i>Ira</i>	<i>Felicidad</i>	<i>Tristeza</i>	<i>Miedo</i>	<i>Disgusto</i>
<i>Veloc. Habla</i>	Ligeramente acelerada	Acelerada o retardada	Pausada	Muy acelerada	Mucho más acelerada
<i>Calidad voz</i>	Procedente del pecho	Estridente	Resonante	Irregular	Retumbante
<i>Intensidad</i>	Alta	Alta	Baja	Normal	Baja

Ilustración 11: Relación de parámetros de la voz con las emociones principales

Existen tres ejemplos de software desarrollados en España, Argentina y México; el primero de ellos forma parte de la Universidad Politécnica de Madrid y se ha desarrollado mediante un lenguaje de programación llamado RFuzzy [24], no existe en la red una descripción adecuada de este software, por lo que se describirán los otros dos.

El software argentino [25] tuvo muy buenos resultados en la clasificación de emociones: la información disponible subraya que obtuvieron un margen de acierto del 76% utilizando las emociones básicas (más una neutral); margen que creció (97%) al discernir entre tres tipos de emociones. Se procesaron un total de 12 características del habla para su posterior evaluación, utilizando un corpus de datos de 535 locuciones (un corpus de la Universidad de Berlín llamado EmoDB, que también figuraba en [23]). Con estos resultados queda de manifiesto la efectividad que tiene el análisis del habla para la extracción de emociones.

En cuanto al software de México [26], es el más reciente y el que tiene una utilidad más directa. Se trata de EmoSpeech, capaz de analizar una conversación e inferir las emociones mediante lo que ellos denominan una “emográfica”, que no es más que una gráfica de las ondas sonoras coloreada en los momentos en los que se produce tensión durante la conversación; esto podría empezar a usarse en los centros de llamadas para detectar una conversación y ver los fallos; de algún modo, servir de entrenamiento para los teleoperadores de cara a los posibles enfados del cliente.

2.5.3. Imagen

Para el análisis de la imagen, clásicamente, se usa una cámara para la extracción de los valores RGB (cantidad de rojo, verde y azul presente en cada píxel de la imagen). Aunque no tienen una representación directa con las emociones, se explican sucintamente algunos parámetros típicos de las imágenes, cuyo conjunto dará lugar a dicha inferencia:

- Resolución: tiene que ver con la cantidad de píxeles (cuadrados que representan un determinado color en la gama RGB, con una intensidad de 8 bits por cada color) que se emplea en la imagen. Al aumentar ésta, aumentará la calidad de la imagen por lo que su análisis será más sencillo, por lo que un requisito determinante en este proyecto es capturar imágenes con una resolución alta, debido a que se espera cierta lejanía por parte del usuario del sistema y una alta resolución implica directamente una mejor detección de sus movimientos (concretamente se usa para el seguimiento o *tracking* de la cara).
- Tono: colores generales de los que está compuesta la imagen. En un proyecto clásico (que no estuviese dotado de un sensor de profundidad para ayudar a identificar el cuerpo) sí habría sido un factor determinante, ya que el reconocimiento de formas viene en buena parte determinado por los cambios de tono (el uso de dos tonos muy similares podría confundir el reconocimiento), pero en este el reconocimiento del cuerpo queda solventado por el sensor de profundidad.
- Brillo: diferente luminosidad u oscuridad que poseen los diferentes tonos. Es importante que la imagen capturada esté en un punto medio o afectará inevitablemente a la eficacia del programa diseñado.
- Contraste: diferencia de la intensidad entre un píxel de la imagen y sus alrededores. En un proyecto clásico también sería muy importante por la misma razón expuesta en el tono (el cambio de contraste también es bastante utilizado en el reconocimiento de formas). En el presente proyecto, aunque la detección del cuerpo no dependa del contraste, sí lo hace la detección facial (un cambio en la resolución de la cámara RGB del Kinect afecta profundamente a una buena o mala detección, por comprobación empírica).

Las aplicaciones en las que se usa para la extracción de emociones tienen que ver normalmente con el estado de ánimo de manera directa (algunos de estos ejemplos están en el [apartado 2.4](#)); sin embargo, existen algunos mecanismos de detección de emociones indirectos que se salen de la norma general de la detección de emociones; un buen ejemplo se halla en [27], un programa capaz de extraer el pulso cardíaco de una persona exclusivamente mediante la webcam. Para ello, se usa un análisis de componentes independiente que divide cada imagen en tres canales distintos: rojo, verde y azul.

Este analizaría la cantidad de colores en lo que denomina “región de interés” (en este caso, la cara del sujeto con el que se experimenta) y sería capaz de determinar cuál es el pulso de la persona a la que está observando.

Un ejemplo de ello se muestra en la Ilustración 12:

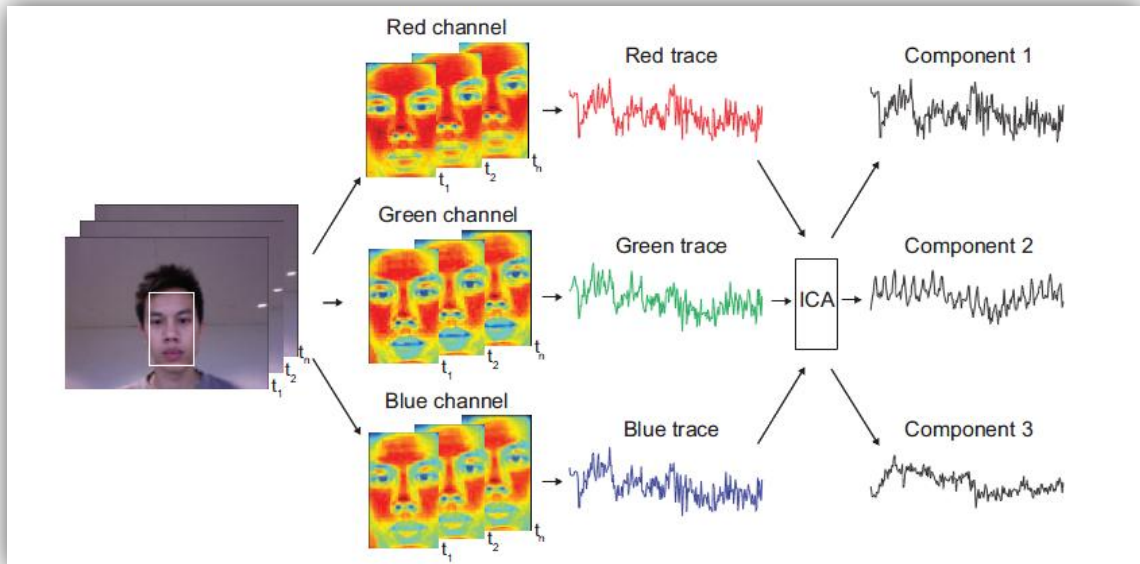


Ilustración 12: Medición del pulso cardíaco mediante cámara

En cuanto a los dispositivos usados para la extracción de emociones, típicamente se usarán siempre cámaras en lugar de algún otro tipo de sensores; hay algunos rivales de *Kinect* muy similares, pero no es el objeto de este proyecto exponerlos.

2.6. Tecnología hardware: *Kinect para Xbox 360* y *Kinect para Windows*

Ambos dispositivos están basado en una tecnología promocionada por la empresa *PrimeSense*: se trata de los tipos de sensores *PrimeSense 3D*. Entre las posibilidades de esta tecnología, cabe destacar dos que aún no se han desarrollado en *Kinect*, y son las siguientes:

- Es capaz de capturar 60 imágenes (*frames*) por segundo, mientras que *Kinect* sólo llega a 30 *frames* por segundo. En el proyecto, esto no es relevante ya que la detección de las emociones no tiene como factor crítico el rango de imágenes capturadas por segundo (se garantiza que por encima de 12 frames por segundo el cambio en detección es muy pequeño).

- Permite según sus notas de prensa capturar el esqueleto (igual que *Kinect*) y también reconocer algunos gestos de las manos (se supone que gracias a las capacidades del sensor de profundidad, lo cual es una ventaja muy a tener en cuenta para proyectos en general). Esta posibilidad queda abierta al uso de un entorno de desarrollo mejorado por parte de Microsoft, o el uso de otro entorno de desarrollo que capacite el reconocimiento de las manos. Hay bastante literatura sobre el tema, además de librerías desarrolladas por los propios usuarios.

Ahora, pasan a exponerse algunas diferencias entre los dispositivos *Kinect para Xbox 360* y *Kinect para Windows* (salido el año de este mismo proyecto), por lo que al mencionar “mejoras” se sobreentiende que se hablará de las del segundo dispositivo respecto al primero:

- Modo cercano establecido en *Kinect para Windows*: permite cercanía de hasta 40 centímetros, mientras que su predecesor (*Kinect para Xbox 360*) no podía extraer información de distancia a menos de 80 centímetros ya que carecía de este modo, como queda de manifiesto en la [Ilustración 13](#):

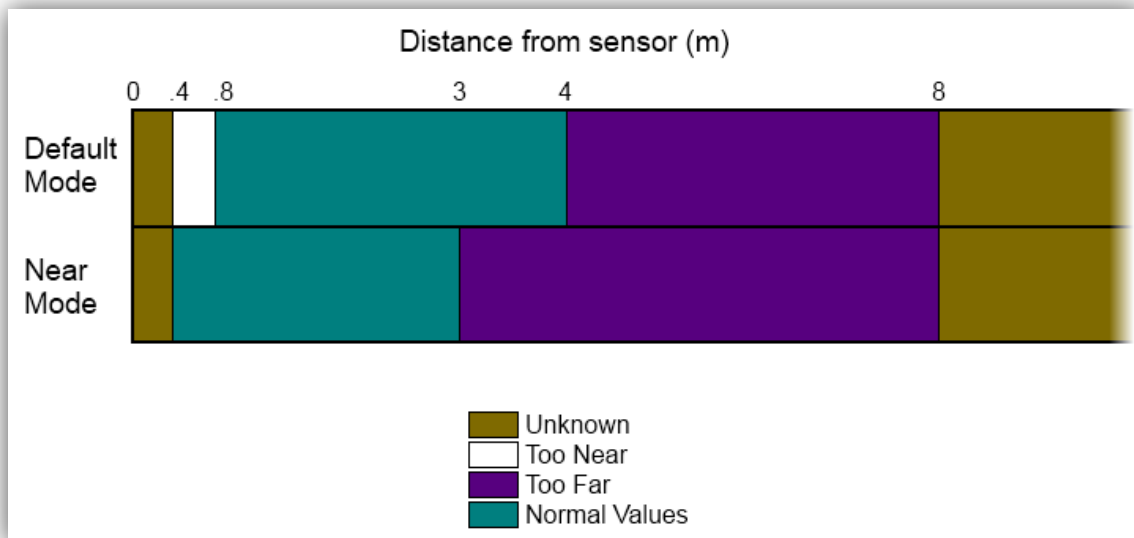


Ilustración 13: Diferencias entre el modo por defecto y el modo cercano de Kinect

- Mejoras en el tiempo de ejecución del dispositivo (a veces *Kinect para Xbox 360* tiene un consumo excesivamente alto de recursos).
- Mejoras en el reconocimiento de voz y de esqueleto (en el primero aumenta la precisión al capturar el sonido, y en el segundo aparte de un aumento de precisión se pueden examinar un menor número de articulaciones, lo que rebaja la carga computacional).

2.7. Especificaciones de *Kinect para Xbox 360*

El dispositivo con el que se va a trabajar presenta las siguientes características:

- Sensores y flujo de datos (*Data Streams*):
 - Cámara de profundidad: 16 bit, 30 *frames* por segundo, resolución 320 x 240.
 - Cámara de color: 32 bit, 30 *frames* por segundo, resolución 640 x 480.
 - Micrófonos: 4 micrófonos con audio de 16 bit, a frecuencia de 16 kilohertzios. Cancelación de eco y reconocimiento de voz múltiple.
 - Los sensores son ajustables con un motor de inclinación que lleva *Kinect* para garantizar el resultado óptimo en el reconocimiento.
- Campo de visión:
 - Horizontal: 57 grados.
 - Vertical: 43 grados.
 - Inclinación física: ± 27 grados.
 - Rango de profundidad del sensor: 1,2 – 3,5 metros.
- Sistema de seguimiento:
 - Rastreo de hasta 6 personas (el rastreo completo del esqueleto sólo se realizará en los usuarios activos).
 - Rastreo de hasta 20 articulaciones por jugador activo.



Ilustración 14: Kinect para Xbox 360

2.8. Entornos de desarrollo para el trabajo con *Kinect*

En este punto se van a exponer brevemente los posibles entornos de desarrollo que podrían usarse para la implementación con *Kinect*. Existen multitud de implementaciones y lenguajes en los que podría haberse implementado, pero con la creación del entorno de desarrollo propio de Microsoft la mayoría quedaría relegada a un segundo plano, siendo los más importantes *OpenNI* y el *SDK* anteriormente mencionado de Microsoft.

2.8.1. *OpenNI*

Este entorno de desarrollo es, según el propio sitio web [30], un entorno de desarrollo utilizado para escribir aplicaciones utilizando la Interacción Natural.

Consiste en un entorno genérico en el cual se comunican los sensores tanto de detección de imagen como de sonido con lo que se denomina *middleware component*, que son componentes software encargados de analizar los datos extraídos del sensor.

La estructura general por capas del entorno *OpenNI* es la que aparece en la Ilustración 15:

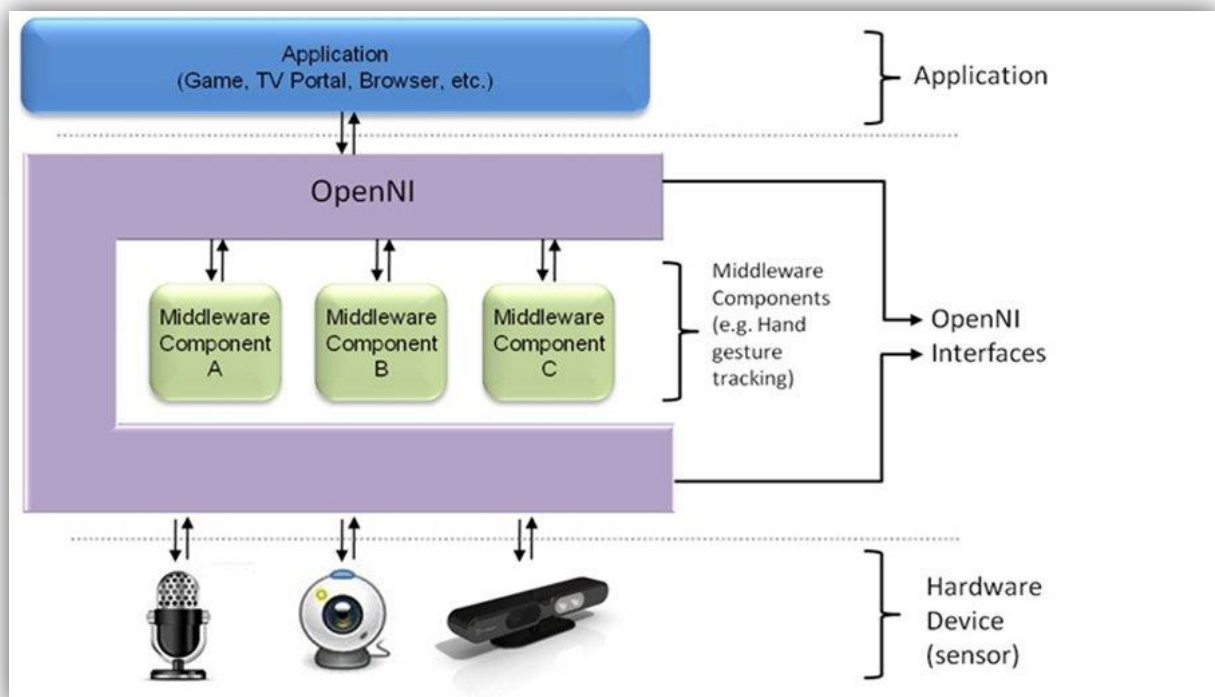


Ilustración 15: Vista por capas del entorno OpenNI

En la capa más baja (sensores) existen soportes para cámaras RGB, infrarrojas, 3D y micrófonos, pero sin duda los componentes más interesantes son las interfaces *OpenNI*, que integran el *middleware* mencionado anteriormente.; entre ellos se puede encontrar:

- Análisis completo del cuerpo: genera información relacionada con el cuerpo (típicamente articulaciones, orientación y centro de masas).
- Análisis del centro de la palma de la mano: proporciona la detección del centro de la palma de la mano, pudiéndose llevar a cabo un análisis de ésta mediante el análisis de convexidades (que nos llevaría a la detección de cada dedo).
- Detección de gestos: componente encargado de identificar gestos predefinidos (como por ejemplo el movimiento de una mano) y realizar una llamada a la aplicación cuando esto ocurra.
- Análisis de escena: analiza la imagen actual para detectar, entre otras cosas, diferentes individuos o las coordenadas del suelo.

Por lo expuesto anteriormente, se considera que *OpenNI* es un entorno muy robusto y útil para aplicaciones muy heterogéneas, aunque el inconveniente principal que presenta es que al ampliar el nivel de abstracción la complejidad de codificación crece; sin embargo, antes de la existencia de un entorno de desarrollo específico para *Kinect* habría sido una muy buena alternativa, ya que multitud de desarrolladores se volcaron en la implementación de todo tipo de programas mediante esta tecnología.

2.8.2. *Kinect para Windows SDK*

Como aporte de Microsoft, surgió junto con su nuevo sensor mejorado al anterior (*Kinect para Windows*) un nuevo entorno de desarrollo fuertemente impulsado por la compañía y la comunidad de desarrolladores. Ejemplos de esta afirmación se encontrarían en la web de Channel9 [31] en la cual aparecen vídeos explicativos para el comienzo del desarrollo con este entorno.

En general tras haber revisado esta tecnología, y elegirla como parte de la solución, destacaría por una parte la virtud del soporte del equipo de Microsoft, que cuenta con una base de conocimiento muy amplia y con profesionales volcados en el desarrollo; por otra parte, sí existe un defecto y es el cambio abrupto entre versiones de este entorno de desarrollo, en el que Microsoft no ha mantenido la retrocompatibilidad en la evolución de la versión 1.0 a la 1.5 (última creada en el cierre de este documento), lo cual debería ser revisado en futuros desarrollos de la misma.

De manera análoga a lo descrito en *OpenNI*, se presentará la arquitectura por capas de este entorno de desarrollo en la Ilustración 16:

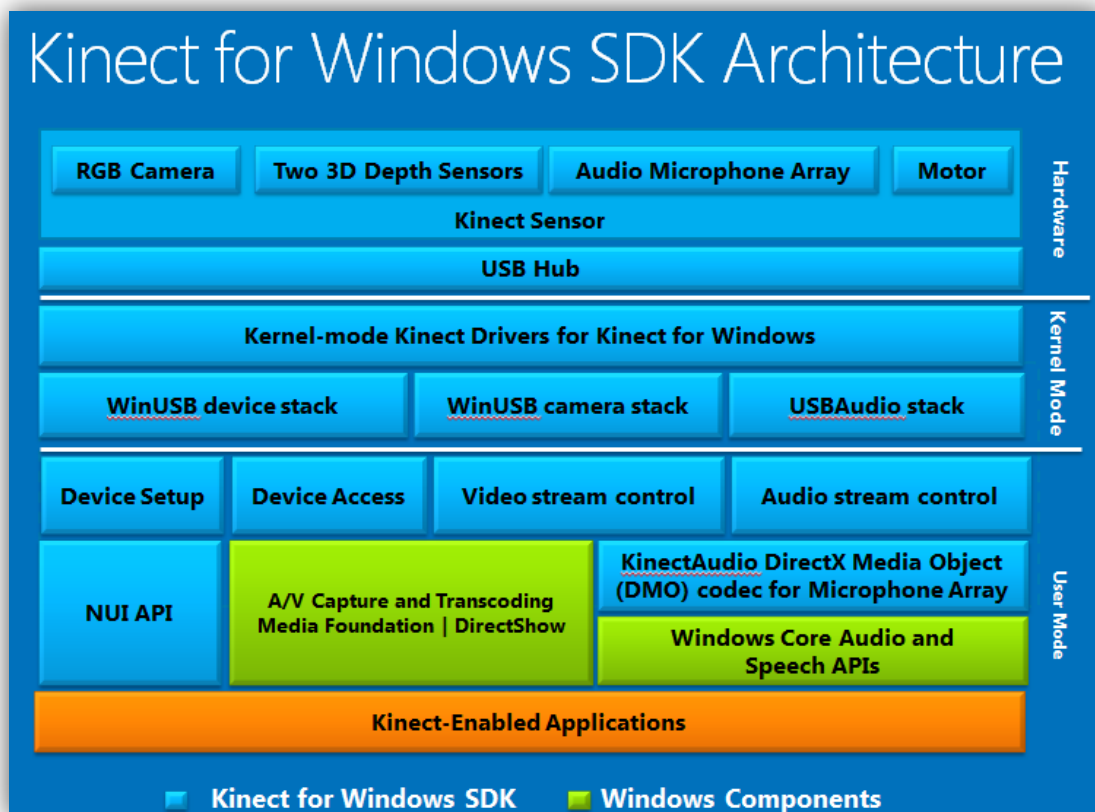


Ilustración 16: Vista por capas del entorno Kinect for Windows SDK

En este caso, a partir de los datos extraídos por el sensor, el entorno de desarrollo ofrece funcionalidades análogas al *OpenNI* en su versión 1.0, y mejoradas ampliamente en su versión 1.5 gracias al análisis facial. A continuación se describen algunas de ellas, realmente útiles para la solución del problema planteado en este proyecto:

- **Análisis del cuerpo:** de manera análoga a *OpenNI*, también ofrece detección de cuerpos; en este caso, puede detectar hasta 6 personas y las articulaciones al completo de hasta 2 personas, aparte de su orientación y su profundidad (lo cual es muy importante para observar los movimientos realizados).
- **No dispone por sí solo de un detector de gestos;** es capaz de detectar algunas poses adoptadas por el usuario mediante el análisis de la situación espacial de cada una de sus articulaciones; esto puede ser analizado mediante la comparación de varias imágenes y observando la evolución de esta orientación espacial.
- **Detector facial:** esto cobra una gran importancia en una aplicación que detecte el estado de ánimo, ya que buena parte de las emociones tienen su máxima expresión en la cara. A diferencia del detector gestual, sí ofrece una detección facial completa.

3. Análisis del sistema

En este punto se va a realizar un análisis del sistema según el problema planteado; para ello, previamente se definirá el problema, para más tarde proceder a focalizar en su resolución por el propio sistema.

La estructura de este punto sería la siguiente:

- Definición del problema.
- Definición del sistema.
- Clasificación de emociones según los gestos del usuario.
- Establecimiento de requisitos.
- Casos de uso.
- Definición de interfaces de usuario. Alternativa escogida.

Esta estructura está basada parcialmente en las partes consideradas más apropiadas para su desarrollo de la Metodología de Sistemas de Información Métrica V3 [32].

3.1. Definición del problema

De manera sintética, el problema principal planteado sería el siguiente: conseguir realizar un detector de emociones mediante el dispositivo *Kinect* que se base para dichas detecciones en el reconocimiento de movimientos relacionados con el cuerpo, los gestos o posturas adquiridos por el mismo y los gestos realizados a nivel facial.

Este análisis y la elección del dispositivo *Kinect* no es trivial: su elección está condicionado por la existencia de un entorno de desarrollo (*SDK* por sus siglas en inglés) de Microsoft que permitía centrarse con profundidad en el problema en lugar de acotar otros previos como la detección de esqueleto y el reconocimiento facial, además de otra característica muy interesante del dispositivo: la posibilidad de captación de imágenes en profundidad, que será de gran utilidad para el desarrollo del proyecto.

Mediante la ilustración 17 queda expresado esquemáticamente cuál es el problema a resolver:

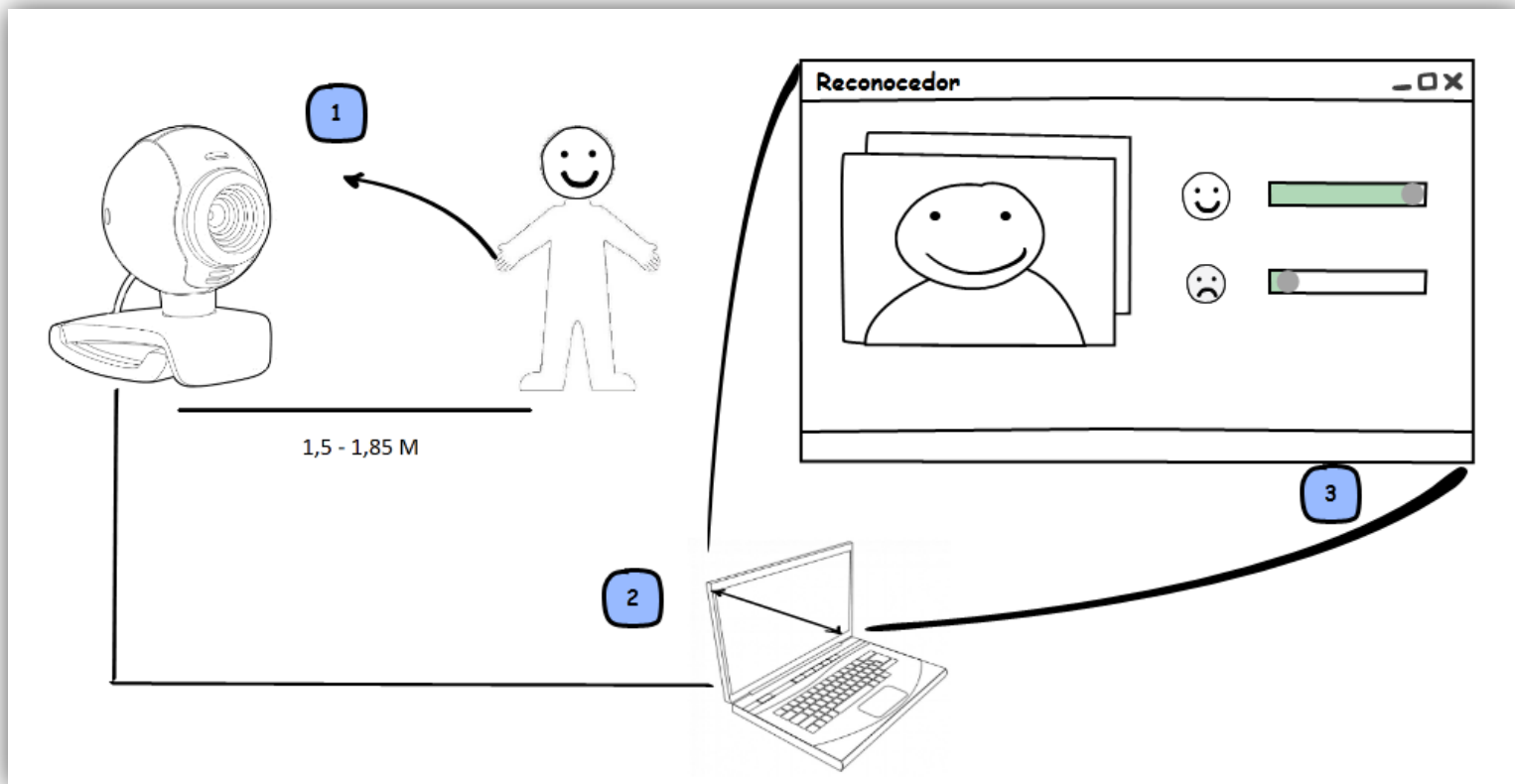


Ilustración 17: Esquema del problema a resolver

Como se ve, el problema consta de 3 fases:

- 1) Grabación de imágenes mediante el dispositivo *Kinect* (suponiendo al individuo en movimiento).
- 2) Recopilación y procesado de la información extraída por las imágenes.
- 3) Uso de los datos en un motor de inferencias para extraer la emoción actual del usuario.

3.2. Definición del sistema

En este apartado se realizarán varias tareas expuestas en Métrica V3 [32]; a saber:

- Alcance del sistema.
- Identificación del entorno tecnológico.
- Identificación de usuarios participantes y finales.

3.2.1. Alcance del sistema

En esta tarea se pretende determinar las restricciones del sistema que se pretende desarrollar. Normalmente para su realización depende de un pliego previo de requisitos de usuario, que en este caso carecería de sentido dado que no existe un cliente final que haya pedido la aplicación.

Sí que se deberán cumplir las siguientes restricciones generales a la hora de desarrollar el sistema, resultando este análisis pormenorizado en la definición de los requisitos de software:

- El sistema utilizado deberá coincidir con el especificado en el entorno tecnológico (apartado 3.2.2) o ser superior al mismo para garantizar su funcionamiento.
- Deberá ajustarse a los requisitos de software definidos en el apartado 3.3.
- El sistema garantizará su funcionamiento en entornos Windows siempre que se cumpla lo especificado en el anexo mencionado.
- Será en la medida de lo posible completamente intuitivo a los usuarios, y transparente a los interesados que gestionen los datos.
- La información presentada en la pantalla debe ser lo más completa y exacta posible.
- No se garantizará en ningún caso su funcionamiento en entornos no Windows o en sistemas inferiores, cabiendo la posibilidad de que pueda ser operacional.
- El sistema tendrá un fuerte enfoque en la recopilación de datos del usuario, por lo que debe garantizarse su buen funcionamiento hasta el punto en el que el usuario termine con la interacción.

3.2.2. Identificación del entorno tecnológico

En este punto se analizarán dos tipos de recursos: físicos y lógicos, cuya implicación económica será posteriormente expuesta en el Anexo C: Presupuesto.

- Recursos físicos:
 - Conexión internet de alta velocidad, 5 Mb/s.
 - Equipo *Fujitsu Amilo Pi 3540*, dotado de procesador *Intel Core 2 Duo P8400*, 4GB de RAM y tarjeta gráfica *NVIDIA GeForce 9300M GS*.
 - Sensor *Kinect para Xbox 360*.
- Recursos lógicos:
 - Lenguaje de programación: *C# 4.0*.
 - Entorno de desarrollo integrado (IDE): *Visual Studio Professional 2010*.
 - Entorno de desarrollo (SDK): *Kinect for Windows SDK 1.5*.
 - Sistema operativo: *Windows 7*, ya incluida su licencia con el equipo.
 - *Microsoft Office 2010*: paquete ofimático con funcionalidades como documentos, hoja de cálculo, presentaciones.
 - Modelado UML: *Altova UModel 2012 Enterprise Edition*, herramienta que ofrece una funcionalidad muy completa para el modelado del sistema.

3.2.3. Identificación de usuarios participantes y finales

Aquí se exponen los distintos usuarios que pueden tener interés en el sistema:

- Investigadores y desarrolladores en el área de *Affective Computing*: su finalidad es la de aportar conocimiento y avances al perfeccionamiento en la detección de emociones, pudiendo además lograr cierto prestigio por sus publicaciones.
- Universidad Carlos III de Madrid: recoge este proyecto de manera que pueda ser consultado en un futuro por alumnos que investiguen en la misma área.
- Usuarios finales: personas que usarán este proyecto para inferencia de emociones o simplemente para la extracción de parámetros relacionados con el movimiento.

3.3. Clasificación de emociones según los gestos del usuario

Este apartado ha sido de importancia crítica, ya que aunque existían numerosas alternativas para el reconocimiento facial como se mostró en el [Apartado 2](#) (Estado del arte), la definición era más imprecisa para el análisis de los gestos corporales relacionados con las emociones.

El análisis y clasificación realizados se han llevado a cabo del siguiente modo:

- 1) Lectura y síntesis de un texto dedicado a la expresión de emociones mediante lenguaje corporal [33]: en él aparece una relación entre las emociones y los cambios principales que aparecen en la postura del cuerpo que han servido como base para el desarrollo de este proyecto. Debe tenerse en cuenta que el objetivo de este texto era el de reproducir emociones creíbles mediante gestos en la creación de avatares virtuales, por lo que algunas de ellas estaban algo sobreactuadas y han sufrido leves modificaciones para conseguir el objetivo de este proyecto: inferir emociones a través de gestos realizados por el usuario.
- 2) Síntesis y profundización en las emociones expresadas por los avatares mediante la lectura del documento referenciado en [34]: diferenciación entre posturas (estáticas y que se mantienen en el sujeto observado) y gestos (dinámicos, que siguen una evolución por parte del sujeto).
- 3) Aporte personal de lo aprendido para la identificación de modo completo y creíble de los gestos realizados y su posible derivación en emociones expresadas por el usuario.
- 4) Realización del [Apartado 2](#) en la que se han detectado las pautas importantes a reconocer en el reconocimiento facial: principalmente, la existencia de las llamadas unidades de animación facial (“AU” por sus siglas en inglés) originarias del modelo Candide3 (mencionado previamente en el [Apartado 2](#) y en [16]) y su presencia en el entorno de desarrollo utilizado (*Kinect for Windows SDK 1.5* [35]). Estas AU son desviaciones de la postura normal que adopta la cara en su reconocimiento, expresando por ejemplo movimientos de la mandíbula o de los extremos de los labios). Posterior aporte personal en la identificación facial de las emociones.
- 5) Acotación del problema de detección de las seis emociones básicas propuestas por Paul Ekman (expuestas previamente en el [Apartado 2.2](#) y manifestada su utilidad según el documento [6]) en un total de cuatro emociones, debido a los problemas de detectar la cólera (diferencias en la frecuencia de movimientos no perceptibles debido a la lejanía con la que se realizan las mediciones) y la diferencia mínima a nivel gestual entre la sorpresa y el miedo (mínima y sólo diferenciable en gestos muy concretos de la cara, como se expresa en [36]).

- 6) Definición de los posibles estados de cada parte corporal (tanto posturas como gestos), tal y como se expresa en la Tabla 1:

Parte corporal	Posibles estados
Cabeza	Erguida, al frente, agachada
Espalda	Atrás, normal, adelantada
Hombros	Encogidos, al frente, atrás
Brazos	Tapando cara, arriba, sueltos, atrás
Boca	Sonrisa, abierta, cerrada
Cejas	Arriba, centro, abajo

Tabla 1: Definición de partes corporales implicadas y posibles estados

- 7) Especificación definitiva del análisis para el posterior diseño de la aplicación de la detección de cuatro emociones: aversión, sorpresa, alegría y tristeza, cuyos rasgos diferenciadores se encuentran expuestos en la Tabla 2. Se deben hacer las siguientes consideraciones sobre la figura:

- Los estados detectados en la boca y las cejas sólo serán relevantes en caso de que el usuario no se encuentre tapando su cara.
- El uso del guión (-) significa que esta parte del cuerpo se ha considerado irrelevante a la hora de hacer la detección para la emoción concreta.

	Aversión	Sorpresa	Alegría	Tristeza
Cabeza	Erguida	Al frente	Erguida /al frente	Agachada
Brazos	-	Arriba / tapando cara	Arriba	Tapando cara / atrás / sueltos
Espalda	Atrás	-	Normal / Atrás	Adelantada
Hombros	Atrás / encogidos	Atrás / encogidos	-	-
Boca	-	Abierta	Sonrisa	Cerrada
Cejas	Abajo	Arriba	Arriba	-

Tabla 2: Clasificación final de las emociones respecto a los gestos detectados

3.4. Establecimiento de requisitos software

Un requisito de software es una especificación de una parte del sistema. Juntando todos los requisitos software se obtiene una descripción completa del comportamiento del sistema que se va a desarrollar. Éstos se dividirán en las categorías siguientes:

- Funcionales: recogen toda la funcionalidad del sistema, definiendo todo el comportamiento del software.
- No funcionales: incluyen criterios que debe cumplir el software a nivel operacional (sus restricciones). Dentro de esta categoría se ha decidido incluir los siguientes cuatro tipos de requisito: rendimiento, operación, interfaz y construcción.
 - De rendimiento: restricciones que debe cumplir la aplicación.
 - De operación: operaciones a realizar con la funcionalidad del sistema.
 - De interfaz: definen elementos de interacción con el usuario.
 - De construcción: directrices a seguir para la construcción del sistema.

Los campos analizados en los requisitos son los siguientes:

- Necesidad: mide lo importante que es un requisito para que funcione el sistema con corrección. Tomará los valores: opcional, conveniente o prescindible.
- Estabilidad: nos da una idea de la probabilidad de que el requisito no cambie en el resto del desarrollo (o en futuros desarrollos). Podrá ser alta, media o baja.
- Verificabilidad: un requisito verificable es aquel para el cual existe un proceso no excesivamente costoso por el cual una persona o máquina puede comprobar que el software lo satisface. Puede ser alta, media o baja.

3.4.1. Requisitos funcionales

RS-01			
Título: Mostrar imagen del usuario			
DESCRIPCION			
El sistema debe mostrar al usuario su imagen mientras esté siendo grabado, en formato RGB, para que éste tenga algo de respuesta o <i>feedback</i> del sistema.			
CARACTERISTICAS			
Necesidad:	<input type="checkbox"/> Opcional	<input checked="" type="checkbox"/> Conveniente	<input type="checkbox"/> Imprescindible
Estabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

Tabla 3: RS-01 Mostrar imagen del usuario

RS-02			
Título: Mostrar el porcentaje de cada emoción			
DESCRIPCION			
En la interacción con el sistema, se debe mostrar el porcentaje de cada una de las emociones que se hayan expresado; serían 4: Alegría, tristeza, aversión y sorpresa. La estabilidad debe ser media sujeta a posibles ampliaciones del rango de las emociones, y se acompañarán estos porcentajes con una barra de progreso.			
CARACTERISTICAS			
Necesidad:	<input type="checkbox"/> Opcional	<input type="checkbox"/> Conveniente	<input checked="" type="checkbox"/> Imprescindible
Estabilidad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

Tabla 4: RS-02 Mostrar el porcentaje de cada emoción

RS-03			
Título: Mostrar el reconocimiento corporal y facial			
DESCRIPCION			
<p>En la interacción con el sistema, se debe mostrar el estado de cada una de las partes que se han considerado implicadas en el cambio emocional; a saber: cabeza, hombros, espalda, brazos, boca y cejas, de manera que el usuario pueda precisar la exactitud de esta información.</p> <p>Esta información se mantendrá expuesta en un rango de entre 1 y 3 segundos si es un gesto que conste de varios movimientos.</p>			
CARACTERISTICAS			
Necesidad:	<input type="checkbox"/> Opcional	<input type="checkbox"/> Conveniente	<input checked="" type="checkbox"/> Imprescindible
Estabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

Tabla 5: RS-03 Mostrar el reconocimiento corporal y facial

RS-04			
Título: Iniciar interacción con el usuario			
DESCRIPCION			
<p>El sistema debe comenzar su interacción con el usuario en el momento en el que detecte su esqueleto mediante <i>Kinect</i>.</p>			
CARACTERISTICAS			
Necesidad:	<input type="checkbox"/> Opcional	<input checked="" type="checkbox"/> Conveniente	<input type="checkbox"/> Imprescindible
Estabilidad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

Tabla 6: RS-04 Iniciar interacción con el usuario

RS-05			
Título: Generar XML con movimientos del usuario			
DESCRIPCIÓN			
<p>En el desarrollo de la interacción con el usuario, el sistema debe ser de generar sus movimientos en un archivo <i>XML</i>, teniendo en cuenta las siguientes restricciones:</p> <ul style="list-style-type: none"> ○ La monitorización debe realizarse en un intervalo entre 1 y 5 segundos. ○ Se deben tener en cuenta el estado de: cabeza, hombros, espalda, brazos, boca y cejas. ○ El formato debe ser el adecuado según el estándar XML. ○ El archivo guardado tendrá la siguiente notación: <i>XMLTrack_dd_mm_aaaa xxhxxmxxs</i>, significando cada parámetro: día, mes, año, horas, minutos y segundos respectivamente. 			
CARACTERÍSTICAS			
Necesidad:	<input type="checkbox"/> Opcional	<input type="checkbox"/> Conveniente	<input checked="" type="checkbox"/> Imprescindible
Estabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja

Tabla 7: RS-01 Generar XML con movimientos del usuario.

RS-06			
Título: Finalizar la generación del XML con movimientos del usuario			
DESCRIPCIÓN			
<p>El sistema dará por finalizada la generación XML con el usuario en el momento en el que deje de ver su esqueleto. Sí mostrará información sobre las emociones y el estado de cada parte del usuario mientras dure la interacción con el mismo.</p>			
CARACTERÍSTICAS			
Necesidad:	<input type="checkbox"/> Opcional	<input checked="" type="checkbox"/> Conveniente	<input type="checkbox"/> Imprescindible
Estabilidad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

Tabla 8: RS-06 Finalizar la generación del XML con movimientos del usuario

RS-07			
Título: Mostrar fiabilidad de medición			
DESCRIPCIÓN			
<p>Para que el usuario conozca en todo momento el estado actual del reconocimiento que se realiza, el sistema deberá ofrecer <i>feedback</i> sobre la fiabilidad de medición, con tres valores distintos basados en un indicador (un icono):</p> <ul style="list-style-type: none"> ○ Indicador positivo: si la distancia del usuario respecto al <i>Kinect</i> está entre 1,6 y 1,75 metros (zona de mejor detección). ○ Indicador medio: si la distancia del usuario respecto a la <i>Kinect</i> oscila entre 1,5-1,6m o entre 1,75-1,85m. ○ Indicador negativo: distancias del usuario respecto a <i>Kinect</i> menores de 1,5 metros o mayores de 1,85 metros (zona de mala detección). 			
CARACTERÍSTICAS			
Necesidad:	<input type="checkbox"/> Opcional	<input type="checkbox"/> Conveniente	<input checked="" type="checkbox"/> Imprescindible
Estabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

Tabla 9: RS-07 Mostrar fiabilidad de medición

RS-08			
Título: Mostrar emoción global del usuario			
DESCRIPCIÓN			
<p>El sistema almacenará el resultado de varias iteraciones con el usuario para así poder inferir una emoción global y mostrársela al usuario. Si ninguna de las emociones (aversión, sorpresa, alegría, tristeza) es mayor o igual del 50% en las últimas iteraciones se mostrará “ninguna” en la interfaz.</p>			
CARACTERÍSTICAS			
Necesidad:	<input type="checkbox"/> Opcional	<input checked="" type="checkbox"/> Conveniente	<input type="checkbox"/> Imprescindible
Estabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

Tabla 10: RS-08 Mostrar emoción global del usuario

3.4.2. Requisitos no funcionales

En estos requisitos se ha decidido introducir un campo adicional llamado “Tipo”, que define el tipo de requisito no funcional según la clasificación hecha al inicio del [Apartado 3.4.](#)

RS-09			
Título: Tiempo de respuesta del sistema			Tipo: Rendimiento
DESCRIPCIÓN			
El tiempo de respuesta del sistema frente a las interacciones realizadas por el usuario mientras haya reconocimiento de esqueleto será menor a un segundo.			
CARACTERÍSTICAS			
Necesidad:	<input type="checkbox"/> Opcional	<input checked="" type="checkbox"/> Conveniente	<input type="checkbox"/> Imprescindible
Estabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

Tabla 11: RS-09 Tiempo de respuesta del sistema

RS-10			
Título: Número de usuarios simultáneos			Tipo: Rendimiento
DESCRIPCIÓN			
El sistema estará capacitado para soportar un solo usuario en cada interacción			
CARACTERÍSTICAS			
Necesidad:	<input type="checkbox"/> Opcional	<input type="checkbox"/> Conveniente	<input checked="" type="checkbox"/> Imprescindible
Estabilidad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

Tabla 12: RS-10 Número de usuarios simultáneos

RS-11			
Título: Especificación de algoritmos de la aplicación			Tipo: Operación
DESCRIPCIÓN			
El sistema se basará en los algoritmos basados en los patrones de la <u>Tabla 2: Clasificación de emociones</u> para la inferencia de emociones a través de gestos.			
CARACTERÍSTICAS			
Necesidad:	<input type="checkbox"/> Opcional	<input type="checkbox"/> Conveniente	<input checked="" type="checkbox"/> Imprescindible
Estabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

Tabla 13: RS-11 Especificación de algoritmos de la aplicación

RS-12			
Título: Interfaz intuitiva			Tipo: Interfaz
DESCRIPCIÓN			
La interfaz será completamente intuitiva y, en la medida de lo posible, coincidir con el modelo mental del usuario. Para ello, se agruparán claramente en el diseño final de la interfaz las siguientes partes del sistema: <ul style="list-style-type: none"> ○ La parte destinada a la inferencia de emociones. ○ Aquella destinada al reconocimiento corporal. ○ Partes que no tengan lugar en la anterior clasificación. 			
CARACTERÍSTICAS			
Necesidad:	<input type="checkbox"/> Opcional	<input checked="" type="checkbox"/> Conveniente	<input type="checkbox"/> Imprescindible
Estabilidad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input type="checkbox"/> Media	<input checked="" type="checkbox"/> Baja

Tabla 14: RS-12 Interfaz intuitiva

RS-13			
Título: Resoluciones de la aplicación		Tipo: Interfaz	
DESCRIPCIÓN			
La interfaz del sistema será vista de la forma ideal siempre que la resolución sea mayor de 897 píxeles de ancho por 568 de alto.			
CARACTERÍSTICAS			
Necesidad:	<input checked="" type="checkbox"/> Opcional	<input type="checkbox"/> Conveniente	<input type="checkbox"/> Imprescindible
Estabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

Tabla 15: RS-13 Resoluciones de la aplicación

RS-14			
Título: Lenguaje de la aplicación			Tipo: Construcción
DESCRIPCIÓN			
El lenguaje usado por la aplicación será C# 4.0, por lo que no se garantiza su funcionamiento en versiones inferiores del lenguaje o en otros lenguajes compatibles con el entorno .NET.			
CARACTERÍSTICAS			
Necesidad:	<input type="checkbox"/> Opcional	<input checked="" type="checkbox"/> Conveniente	<input type="checkbox"/> Imprescindible
Estabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja

Tabla 16: RS-14 Lenguaje de la aplicación

RS-15			
Título: Librerías usadas para <i>Kinect</i>			Tipo: Construcción
DESCRIPCIÓN			
La librería empleada para el desarrollo del sistema será <i>Kinect for Windows SDK 1.5</i> , sin poder garantizarse su funcionamiento en versiones anteriores o posteriores de la misma			
CARACTERÍSTICAS			
Necesidad:	<input type="checkbox"/> Opcional	<input type="checkbox"/> Conveniente	<input checked="" type="checkbox"/> Imprescindible
Estabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Verificabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

Tabla 17: RS-15 Librerías usadas para Kinect

3.5. Casos de uso

Un caso de uso es una descripción de los pasos o actividades que deben realizarse para llevar a cabo un proceso del sistema. Muestra así la interacción entre un actor y el sistema que se va a desarrollar, con la finalidad de facilitar el entendimiento de cómo funcionan los procesos internamente.

Los campos analizados en los casos de uso serían los siguientes:

- Actores: quién desempeña las acciones del caso de uso.
- Objetivo: descripción breve de la finalidad del caso de uso.
- Precondiciones: en qué estado tiene que estar el sistema para poder llevarse a cabo el caso de uso.
- Postcondiciones: en qué estado queda el sistema tras realizar el caso de uso de forma normal para su escenario básico.
- Escenario básico: secuencia de pasos que hay que seguir, de forma ordenada, en un flujo normal del sistema.

Para este caso concreto, sólo se han identificado dos casos de uso, partiendo del supuesto de que los dos actores presentes pueden observar cómo interactúa el sistema con ellos, pero sólo el observador de la acción (el que está “detrás de la cámara”) tiene interés en evaluar las interacciones previas del observado con el sistema.

A modo ilustrativo, se representará esquemáticamente según la notación UML [37] los casos de uso que se desarrollarán en el sistema (ilustración 18):

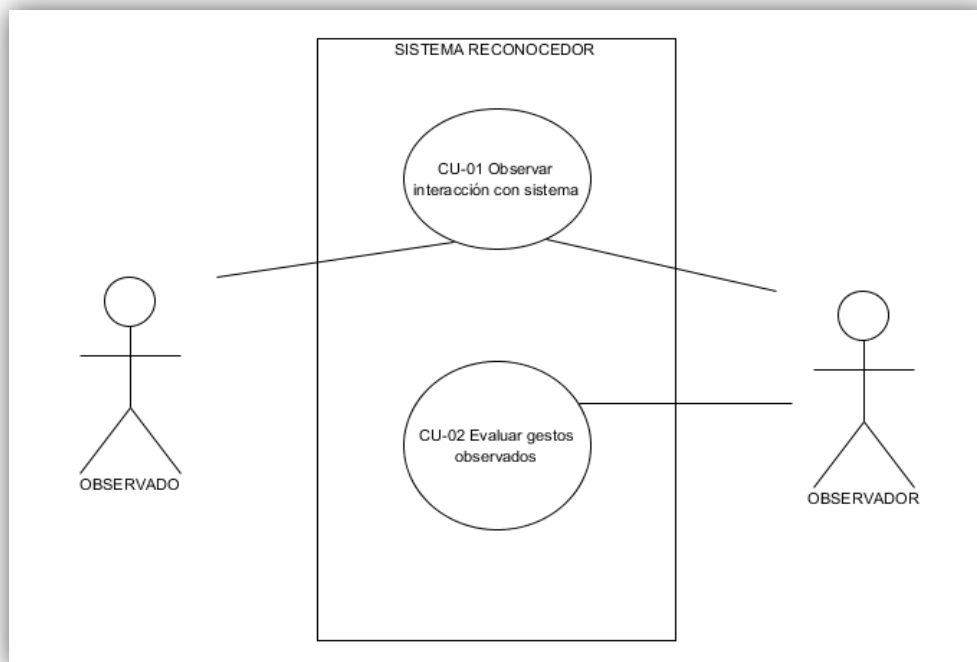


Ilustración 18: Diagrama de Casos de Uso

CU-01	
Título: Observar interacción con sistema	
Actores: Usuario observador, usuario observado.	
Objetivo: El observador (y el observado en caso de que se desee) observará su interacción con el sistema y comprobará su corrección.	
Precondiciones: El esqueleto debe estar debidamente reconocido.	
Postcondiciones: Se almacenará información sobre la interacción del observado.	
ESCENARIO BÁSICO	
<ol style="list-style-type: none"> 1. El observado se coloca frente a la cámara. 2. Cuando la cámara está lista aparece su imagen indicándole que puede comenzar la interacción. 3. El observado sale del ángulo de visión de la cámara, se guarda información de su interacción y ésta se termina. 	

Tabla 18: CU-01 Interactuar con sistema

CU-02	
Título: Evaluar gestos observados	
Actores: Usuario observador	
Objetivo: El observador deberá tener acceso a cualquiera de las interacciones previas de un observado.	
Precondiciones: Se ha realizado al menos una interacción con el observado.	
Postcondiciones: Ninguna.	
ESCENARIO BÁSICO	
<ol style="list-style-type: none"> 1. El observador accede a los documentos XML que registran los movimientos del usuario. 2. Escoge la fecha concreta que le resulte relevante. 3. Termina su interacción. 	

Tabla 19: CU-02 Evaluar gestos observados

3.6. Definición de interfaces de usuario. Alternativa escogida.

Para finalizar el análisis del sistema, se va a mostrar de manera sucinta la evolución que se sigue al analizar las necesidades de la aplicación en cuanto al diseño de su interfaz principal.

La primera alternativa considerada ha sido la siguiente (ilustración 19):

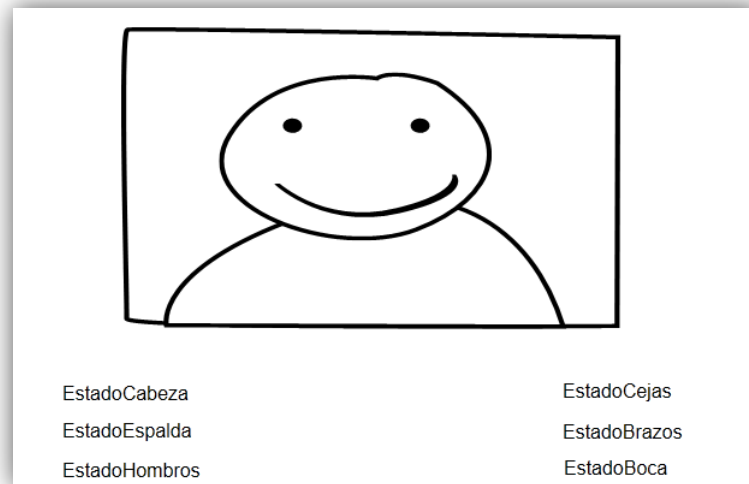


Ilustración 19: Primera alternativa de diseño de la aplicación

Esta alternativa es de gran utilidad para la primera fase en la que se prueba la funcionalidad que da sentido a la aplicación: la detección del estado de cada una de las partes implicadas en la detección de emociones.

Una vez conseguida una detección suficientemente buena, se procede a seguir con la fase de pruebas, lo que daría lugar a un esquema similar al siguiente (ilustración 20):

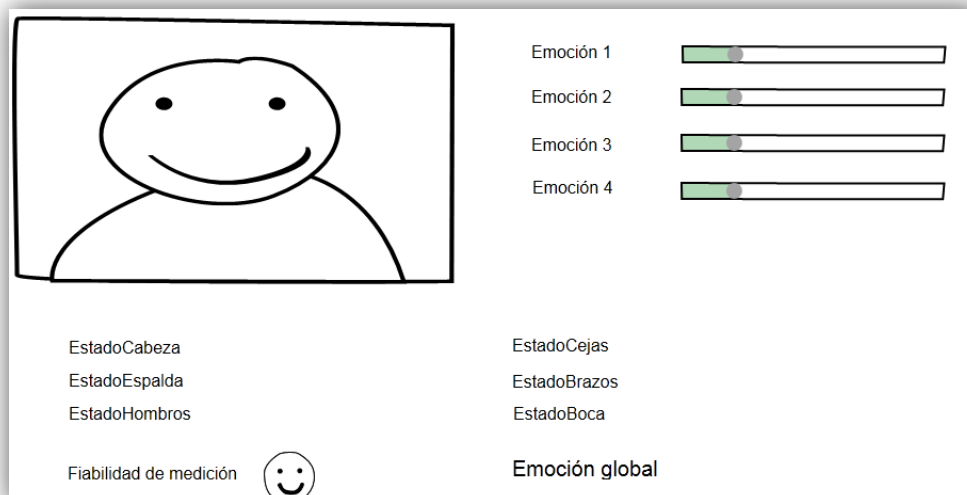


Ilustración 20: Segunda alternativa de diseño de la aplicación

Se observa que esta alternativa es mucho más clara e incorpora elementos muy intuitivos para el usuario; a saber: el uso de barras de progreso como indicador de porcentaje aproximado de las emociones que en ese momento se han detectado, lo que se ha denominado un indicador de fiabilidad (expuesto en el requisito **RS-07** del presente documento) y además un medidor de la emoción global (expuesto en el requisito **RS-08**).

Aunque podría servir de esquema definitivo, si se analiza detalladamente el requisito **RS-12** se aprecia que no termina de satisfacerse, ya que los elementos no están del todo agrupados correctamente y la definición en tiempo real (ya que la interfaz sufre varios cambios por segundo) puede dar lugar a confusiones. Ya que el tiempo de modificación para satisfacer este requisito es razonablemente pequeño, se procede al diseño del tercer y último esquema para la interfaz del usuario, que será el escogido finalmente (ilustración 21):

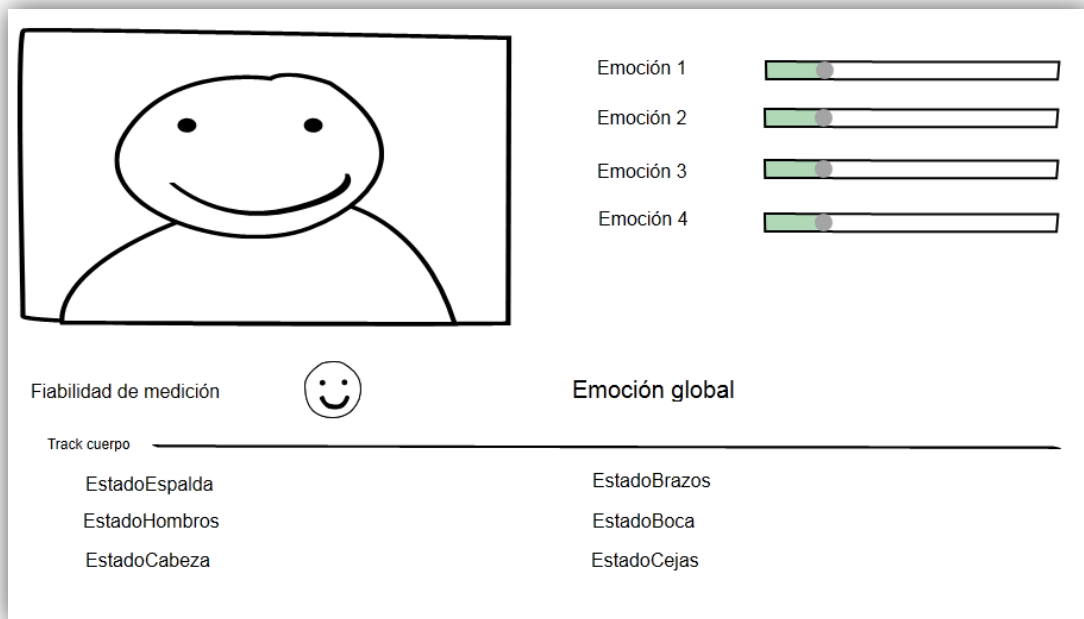


Ilustración 21: Tercera alternativa de diseño de la aplicación

En esta alternativa sí se aprecia claramente la diferenciación de cada una de las partes, y de un solo vistazo podría entenderse perfectamente qué hace la aplicación y qué se espera que haga el usuario, resultando en una interfaz intuitiva y satisfaciendo de manera completa los requisitos de software previamente mencionados.

4. Diseño del sistema

Tras haber definido y mostrado una especificación detallada de lo que debe hacer el sistema y contemplado cómo debería ser el interfaz escogido, en este punto se especificarán las funcionalidades del sistema junto con una exposición de la estructura del mismo (definición de arquitectura del sistema), además de realizarse una revisión de los algoritmos que más importantes han resultado para la resolución del sistema. Por tanto, la estructura de este punto sería como sigue:

- Diagrama de paquetes.
- Análisis de clases.
- Algoritmos usados.
- Interfaz de usuario utilizada.

4.1. Diagrama de paquetes

Dado que se quiere realizar un análisis completo de las clases, se procede previamente a explicar un nivel por encima cómo está dividido este sistema en agrupaciones lógicas. Esto es lo que se conoce como diagrama de paquetes y es interesante para contextualizar la aplicación dentro de la estructura de la que depende. No se pretende hacer un análisis en profundidad de cada paquete sino mostrar la organización de los mismos.

Esta descomposición sería como se muestra en la ilustración 22:

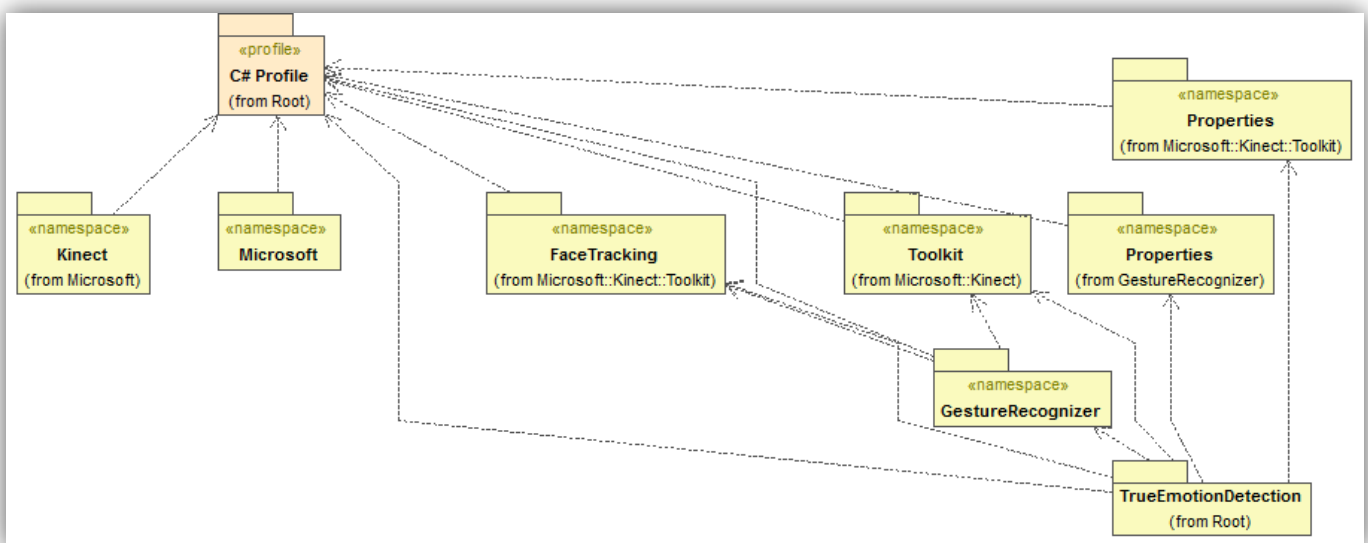


Ilustración 22: Diagrama de paquetes del sistema

De este diagrama de paquetes cabe señalar los siguientes componentes, usando una aproximación hacia abajo, conocida como *top-down* por sus siglas en inglés (primero el paquete principal y luego su descomposición en subpaquetes):

- Paquete *TrueEmotionDetection*: es el paquete que contiene la solución al proyecto completo. Está compuesto de tres subpaquetes: *GestureRecognizer*, *Microsoft.Kinect.Toolkit* y *Microsoft.Kinect.Toolkit.FaceTracking*, que a nivel lógico serán proyectos independientes y serán tratados como componentes distintos, además de contemplar el acceso a las propiedades de estos proyectos.
- Paquete *GestureRecognizer*: en él se encuentra el proyecto implementado, que además deberá comunicarse con los otros proyectos existentes para poder acceder a la *Kinect* y sus mecanismos. Éste paquete será analizado en profundidad en el Apartado 4.2.
- Paquete *Microsoft.Kinect.Toolkit*: proyecto que contiene al componente de Microsoft *KinectSensorChooser*, que en este entorno de desarrollo (SDK 1.5) será el encargado de acceder a la información proporcionada por *Kinect*. Permite manejar el ciclo de vida de *Kinect* en la aplicación (comienzo y fin de detección del mismo) y además proporciona un control (parte de un formulario que muestra datos) llamado *KinectSensorChooserUI* que nos permitirá mostrar directamente en la interfaz de usuario el acceso a los datos en tiempo real.
- Paquete *Microsoft.Kinect.Toolkit.FaceTracking*: proyecto que proporciona acceso a las funcionalidades de la detección facial de este entorno de desarrollo concreto, que son entre otras la deducción de la postura de la cabeza y de las expresiones faciales en tiempo real, información que también debe ser usada para lograr un mejor reconocimiento de emociones.

4.2. Análisis de clases

A continuación se mostrará el diagrama de clases simplificado, para mostrar la relación entre las clases usadas en el proyecto (ilustración 23), que constará de dos partes bien diferenciadas que se analizarán por separado:

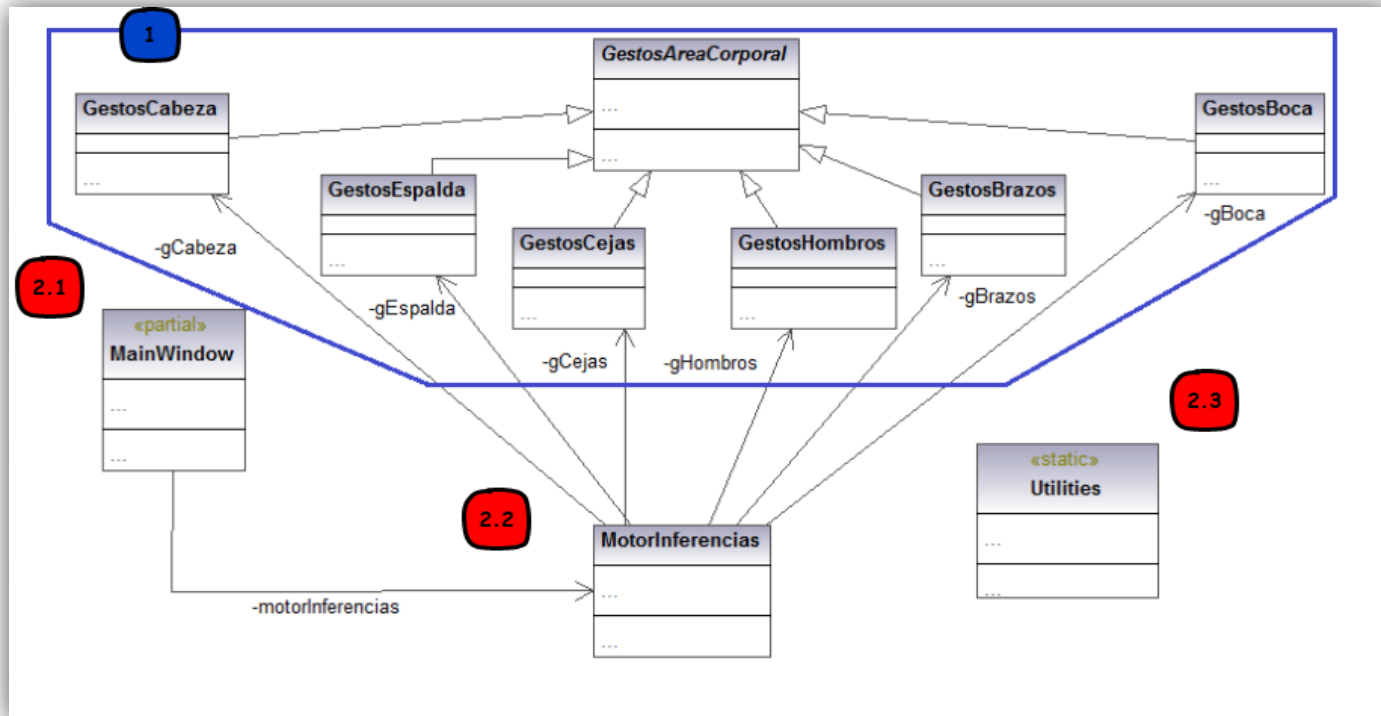


Ilustración 23: Diagrama de clases simplificado

Como se puede observar, el proyecto se basa casi completamente en los gestos, que es donde se ha focalizado el esfuerzo realizado ya que formaba parte de los objetivos principales. Todos los gestos tendrán componentes comunes, y el motor de inferencias (la clase encargada de inferir emociones a través de gestos) extraerá la información en tiempo real sobre cada uno de estas áreas corporales.

La clase *MainWindow* será la responsable de mostrar la interfaz de usuario e interactuar con el *Kinect*, además de escribir al final de la interacción un *XML* con todos los gestos observados.

La clase *Utilities* tendrá algunas utilidades genéricas que se puedan usar por varias clases a lo largo del proyecto, además de aquellos métodos que no estén directamente relacionados con la extracción de gestos o la inferencia de emociones (por ejemplo, escribir un *XML* con la información sobre la interacción actual).

Una vez examinado este esquema general, se mostrará el esquema detallado de clases (con atributos y métodos inclusive) en la ilustración 24:

Una vez mostrado el esquema en su conjunto, se va a proceder a analizar el diagrama de clases por partes según lo mostrado anteriormente en la ilustración 23.

El identificador usado para las clases será Clase del Modelo Detallado – Número (CMD-XX). Los campos contemplados para el análisis de cada clase son los siguientes:

- Nombre de clase: nombre de la clase tal y como va a implementarse.
- Tipo de la clase: indica los modificadores de la clase (si los tiene).
- Interfaces implementadas: indica las posibles interfaces que implementa.
- Clase padre: muestra si la clase hereda de alguna otra.
- Dependencias: muestra si esta clase depende de alguna otra.
- Responsabilidad: breve síntesis sobre la funcionalidad de la clase.
- Structs (si tiene alguna estructura).
- Enum (si tiene algún enumerado).
- Atributos.
- Operaciones.

4.2.1. Primera parte del diagrama de clases

Esta parte se correspondería con la parte rodeada de azul y marcada con un 1 en la ilustración 23, y comprende toda la parte de la lógica relacionada con la detección de gestos corporales y faciales. A continuación se procede a mostrar tablas explicativas de cada una de sus clases:

CMD-01	
Nombre de clase: GestosAreaCorporal	Tipo de clase: Abstracta
Interfaces implementadas: Ninguna	Clase padre: Ninguna
Dependencias: Ninguna	
Responsabilidad: Proporciona la base para la detección de gestos del área corporal	
ESTRUCTURAS	

- **PosturasEnTiempo**: estructura que almacena una lista de posturas y otra de fechas que funciona de nexo entre ambas.
- **Vector4**: estructura que contiene una lista de posiciones (X, Y, Z) y una fecha en la que son calculadas.
- **PropiedadesGesto**: propiedades que tienen los gestos que se prolongan en el tiempo: altura, anchura, profundidad, duración mínima del gesto, duración máxima del gesto y un parámetro que indica el avance en altura, anchura o profundidad que se produce en cada iteración en el gesto concreto.

ENUMERADOS

- **Posture**: enumerado con todas las posibles posturas y gestos que podemos obtener de la interacción teniendo en cuenta las principales áreas del cuerpo; está compuesto por los valores expresados en la siguiente tabla (que apareció en la fase de análisis previamente):

Parte corporal	Posibles estados
Cabeza	Erguida, al frente, agachada
Espalda	Atrás, normal, adelantada
Hombros	Encogidos, al frente, atrás
Brazos	Tapando cara, arriba, sueltos, atrás
Boca	Sonrisa, abierta, cerrada
Cejas	Arriba, centro, abajo

ATRIBUTOS

- **IList<Posture>** posturasDetectadas: Lista de posturas detectadas en un intervalo definido.
- **Posture** ultimoGestoAreaCorporal: Almacena el último gesto detectado.
- **PosturasEnTiempo** posturasEnTiempo: almacena las posturas en relación a una fecha.

OPERACIONES

- **PosturasEnTiempo** GetPosturasDetectadas(): Devuelve una lista de posturas detectadas con sus respectivas fechas.
- **void** GuardarPose(**Posture** anterior, **DateTime** fecha): Método que almacena la pose actual si ésta es distinta que la anterior pose almacenada.
- **IDictionary<string,float>** RestaDistancias(**Skeleton** skeleton, **String** joint1, **String** joint2): Método que calcula las distancias entre un joint (articulación del esqueleto) dado y otro.
- **Vector4** JointToVector4(**Skeleton** skeleton, **String** joint): Devuelve las coordenadas XYZ y la fecha de la llamada mediante el parámetro *joint*, que indica la articulación concreta que queremos obtener.

Tabla 20: CMD-01 Clase GestosAreaCorporal

CMD-02	
Nombre de clase: GestosCabeza	Tipo de clase: Normal
Interfaces implementadas: Ninguna	Clase padre: GestosAreaCorporal
Dependencias: Ninguna	
Responsabilidad: Proporciona información sobre la postura actual de la cabeza.	
ATRIBUTOS	
Ninguno	
OPERACIONES	
<ul style="list-style-type: none"> ○ bool CabezaAgachada(Skeleton skeleton): Detecta si la cabeza está agachada. ○ bool CabezaErguida(Skeleton skeleton): Detecta si la cabeza está erguida. ○ bool CabezaAlFrente(Skeleton skeleton): Detecta si la cabeza está l frente. 	

Tabla 21: CMD-02 Clase GestosCabeza

CMD-03	
Nombre de clase: GestosEspalda	Tipo de clase: Normal
Interfaces implementadas: Ninguna	Clase padre: GestosAreaCorporal
Dependencias: Ninguna.	
Responsabilidad: Proporciona información sobre la postura actual de la espalda.	
ATRIBUTOS	
Ninguno	
OPERACIONES	
<ul style="list-style-type: none"> ○ bool EspaldaAlFrente(Skeleton skeleton): Detecta si la espalda está al frente. ○ bool EspaldaHaciaDelante(Skeleton skeleton): Detecta si la espalda está hacia delante. ○ bool EspaldaHaciaAtras(Skeleton skeleton): Detecta si la espalda está hacia atrás. 	

Tabla 22: CMD-03 Clase GestosEspalda

CMD-04	
Nombre de clase: GestosHombros	Tipo de clase: Normal
Interfaces implementadas: Ninguna	Clase padre: GestosAreaCorporal
Dependencias: Ninguna	
Responsabilidad: Proporciona información sobre el movimiento de los hombros durante la interacción.	
ATRIBUTOS	
Ninguno	
OPERACIONES	
<ul style="list-style-type: none"> ○ bool HombrosAlFrente(List<Vector4> listaPosiciones): Detecta si los hombros se dirigen al frente (posición normal). ○ bool HombrosEncogidos(List<Vector4> listaPosiciones): Detecta si los hombros están encogiéndose. ○ bool HombrosHaciaAtras(List<Vector4> listaPosiciones): Detecta si los hombros están yendo hacia atrás. 	

Tabla 23: CMD-04 Clase GestosHombros

CMD-05	
Nombre de clase: GestosBrazos	Tipo de clase: Normal
Interfaces implementadas: Ninguna	Clase padre: GestosAreaCorporal
Dependencias: Ninguna	
Responsabilidad: Proporciona información sobre el movimiento de los brazos durante la interacción.	
ATRIBUTOS	
Ninguno	
OPERACIONES	
<ul style="list-style-type: none"> ○ bool BrazosHaciaArriba(List<Vector4> listaPosiciones): detecta si los brazos se dirigen hacia arriba. ○ bool BrazosTapandoCara(Skeleton skeleton): detecta si los brazos están tapando en un momento dado la cara. ○ bool BrazosHaciaAtras(List<Vector4> listaPosiciones): detecta si los brazos se dirigen hacia atrás. ○ bool BrazosSuelos(List<Vector4> listaPosiciones): detecta si los brazos están sueltos. 	

Tabla 24: CMD-05 Clase GestosBrazos

CMD-06	
Nombre de clase: GestosBoca	Tipo de clase: Normal
Interfaces implementadas: Ninguna	Clase padre: GestosAreaCorporal
Dependencias: Ninguna	
Responsabilidad: Proporciona información sobre la desviación de los parámetros de la boca respecto del estado normal.	
ATRIBUTOS	
<i>Ninguno</i>	
OPERACIONES	
<ul style="list-style-type: none"> ○ <code>string</code> detectarBoca(<code>FaceTrackFrame</code> cara): devuelve un string con la forma que ha adquirido la boca actualmente: sonriente, abierta o cerrada. 	

Tabla 25: CMD-06 Clase GestosBoca

CMD-07	
Nombre de clase: GestosCejas	Tipo de clase: Normal
Interfaces implementadas: Ninguna	Clase padre: GestosAreaCorporal
Dependencias: Ninguna	
Responsabilidad: Proporciona información sobre la desviación de los parámetros de las cejas respecto del estado normal.	
ATRIBUTOS	
<i>Ninguno</i>	
OPERACIONES	
<ul style="list-style-type: none"> ○ <code>string</code> detectarCejas(<code>FaceTrackFrame</code> cara): devuelve un string con la posición actual de las cejas: arriba, centradas o abajo. 	

Tabla 26: CMD-07 Clase GestosCejas

4.2.2. Segunda parte del diagrama de clases

Esta parte se correspondería con las clases marcadas por la signatura (en rojo) 2.x de la [Ilustración 23](#). Aquí se englobarían el resto de clases del paquete *GestureRecognizer*, cada una de ellas con una funcionalidad y tareas definidas.

CMD-08	
Nombre de clase: MainWindow	Tipo de clase: Parcial
Interfaces implementadas: Ninguna	Clase padre: Window
Dependencias: MotorInferencias	
Responsabilidad: Proporciona una lógica de interacción entre la interfaz que ve el usuario y las operaciones realizadas internamente.	
ATRIBUTOS	
<ul style="list-style-type: none"> ○ <code>private FaceTracker</code> faceTracker: almacena la imagen actual obtenida de la cara. ○ <code>private KinectSensor</code> newSensor: se encarga de inicializar y destruir la interacción con los componentes de <i>Kinect</i>. ○ <code>private short[]</code> depthPixelData: matriz que almacena los datos de profundidad de una imagen en cada iteración. ○ <code>private byte[]</code> colorPixelData: matriz que almacena los datos RGB de una imagen en cada iteración. ○ <code>private KinectSensorChooser</code> sensorChooser: variable que nos permite escoger una de las partes del sensor <i>Kinect</i>. ○ <code>private MotorInferencias</code> motorInferencias: clase encargada de evaluar la información extraída en cada iteración mediante <i>Kinect</i>. ○ <code>private bool</code> cerrando: da información sobre el cierre del programa, para finalizar la interacción con el dispositivo <i>Kinect</i>. 	
OPERACIONES	
<ul style="list-style-type: none"> ○ <code>public MainWindow()</code>: Constructor de la clase, inicializa variables. ○ <code>void Window_Loaded(object sender, RoutedEventArgs e)</code>: agrega el sensor <i>Kinect</i> como parte de la interacción en una cola de eventos. ○ <code>void sensorChooser_KinectChanged(object sender, KinectChangedEventArgs e)</code>: inicializa el <i>Kinect</i>. ○ <code>void sensor_AllFramesReady(object sender, AllFramesReadyEventArgs e)</code>: método que es llamado en cada iteración: obtiene toda la información necesaria de <i>Kinect</i> y se la proporciona al motor de inferencias. ○ <code>void StopKinectSensorChooser(KinectSensorChooser sensorCh)</code>: detiene el sensor. ○ <code>void Window_Closing(object sender, System.ComponentModel.CancelEventArgs e)</code>: Se accede al cerrar el programa, su cometido es detener el sensor <i>Kinect</i> y escribir un XML con el contenido de la interacción con el usuario. 	

Tabla 27: CMD-08 Clase MainWindow

CMD-09	
Nombre de clase: MotorInferencias	Tipo de clase: Normal
Interfaces implementadas: Ninguna	Clase padre: Ninguna
Dependencias: GestosCabeza, GestosEspalda, GestosHombros, GestosBrazos, GestosBoca, GestosCejas	
Responsabilidad: Inferir una emoción a través de los últimos gestos reconocidos.	
ATRIBUTOS	
<ul style="list-style-type: none"> ○ <code>private GestosCabeza</code> gCabeza: variable con los gestos de la cabeza . ○ <code>private GestosEspalda</code> gEspalda: variable con los gestos de la espalda. ○ <code>private GestosHombros</code> gHombros: variable con los gestos de los hombros. ○ <code>private GestosBrazos</code> gBrazos: variable con los gestos de los brazos. ○ <code>private GestosBoca</code> gBoca: variable con los gestos de la boca. ○ <code>private GestosCejas</code> gCejas: variable con los gestos de las cejas. ○ <code>private IDictionary<string, List<GestosAreaCorporal.Vector4>></code> listaPosiciones: Diccionario que contiene las posiciones (X,Y,Z) y fecha con todos los joints actuales. ○ <code>List<float[]></code> conjuntoEmociones: lista que almacena en cada iteración la última emoción infería para más tarde poder mostrar la medición global. 	
OPERACIONES	
<ul style="list-style-type: none"> ○ <code>public MotorInferencias()</code>: constructor de clase, inicializa variables. ○ <code>public float[]</code> GetPorcentajeEmociones(<code>Skeleton</code> skeleton, <code>FaceTrackFrame</code> faceFrame): método que obtiene el porcentaje de cada emoción en la iteración actual. ○ <code>public int</code> emocionGlobal(<code>int</code> numeroMediciones): método que infiere la emoción global del usuario tras un número de mediciones especificado por parámetro. ○ <code>public void</code> ComprobarPosesActuales(<code>Skeleton</code> skeleton, <code>FaceTrackFrame</code> faceFrame): método que comprueba las posiciones de todas las partes del cuerpo. ○ <code>public float</code> GetPorcentajeAlegria(): obtiene una ponderación del porcentaje de alegría. ○ <code>public float</code> GetPorcentajeTristeza(): obtiene una ponderación del porcentaje de tristeza. ○ <code>public float</code> GetPorcentajeAversion(): obtiene una ponderación del porcentaje de aversión. ○ <code>public float</code> GetPorcentajeSorpresaMiedo(): obtiene una ponderación del porcentaje de sorpresa (aunque el miedo sería prácticamente análogo). ○ <code>public string[]</code> GetPosturasGUI(): método que vuelca a un <i>array</i> de strings las posturas actuales. ○ <code>public void</code> EscribirXML(): agrega todas las poses detectadas en ejecución y escribe un XML con ellas. 	

Tabla 28: CMD-09 Clase MotorInferencias

CMD-10	
Nombre de clase: Utilities	Tipo de clase: Estática
Interfaces implementadas: Ninguna	Clase padre: Ninguna
Dependencias: Ninguna	
Responsabilidad: Proporcionar algunas utilidades genéricas a la aplicación.	
ESTRUCTURAS	
<ul style="list-style-type: none"> ○ PosesPorFecha: struct que relaciona una pose con una fecha y con una parte corporal 	
ATRIBUTOS	
<ul style="list-style-type: none"> ○ const int skeletonCount : Número de esqueletos reconocibles por una aplicación en el entorno. ○ static Skeleton[] allSkeletons : Array con los esqueletos reconocidos. ○ static String[] jointList: lista estática con los joints (articulaciones) reconocidas del esqueleto. 	
OPERACIONES	
<ul style="list-style-type: none"> ○ public static Skeleton GetFirstSkeleton(AllFramesReadyEventArgs e): obtiene el primer esqueleto reconocido de la iteración. ○ public static void AnyadirTodosJoints(IDictionary<string, List<GestosAreaCorporal.Vector4>> di): Añade al diccionario los string que pueden ser accedidos. ○ public static void AnyadirPosicionesActuales(IDictionary<string, List<GestosAreaCorporal.Vector4>> listaPosiciones, GestosAreaCorporal gestos, Skeleton first): método que agrega a una lista de posiciones una nueva posición detectada para que más tarde se pueda comparar con otras previas. ○ public static void LimpiarListaPosiciones(IDictionary<string, List<GestosAreaCorporal.Vector4>> listaPosiciones, int numeroPosiciones): elimina de una lista dinámica de posiciones la primera de ellas. ○ public static LinkedList<String> PruebasDistancia(Skeleton skeleton, String zona1, String zona2): devuelve una lista con la distancia entre las articulaciones especificadas en zona1 y zona2. ○ public static int obtenerFiabilidadMedicion(Skeleton skeleton): devuelve un entero que indica cómo de bien funciona el reconocimiento según la distancia del usuario: 0 si funcionará mal, 1 si funcionará de manera aceptable y 2 si funciona de manera óptima. ○ public static IList<PosesPorFecha> ObtenerTodasPoses(Dictionary<string, GestosAreaCorporal.PosturasEnTiempo> poses): Obtiene todas las poses de todas las partes del cuerpo en la interacción con el usuario. ○ public static void EscribirXmlPoses(string rutaFichero, Dictionary<string, GestosAreaCorporal.PosturasEnTiempo> poses): Escribe un XML con las poses ordenadas por fecha. 	

Tabla 29: CMD-10 Clase Utilities

4.3. Algoritmos usados

Se ha considerado interesante exponer una pequeña parte de la aplicación que contenga los algoritmos utilizados en la detección de gestos, ya que siguen un patrón común que podría ser interesante en futuras aplicaciones, además de dar una idea bastante cercana de lo que se ha diseñado. Además, se expondrá cómo se ha llevado a cabo la detección de gestos para la inferencia de emociones.

4.3.1. Algoritmo de detección de poses

Este tipo de algoritmos serán bastante simples y serán usados en la detección de poses de la cabeza, de la espalda y para la detección de los brazos tapando cara.

Su uso es razonable para la detección de movimientos corporales en las situaciones en que sea posible detectar una pose concreta, entendiéndose por pose una posición fija que toma el esqueleto reconocido; además, es importante que la pose percibida sea extensible a un amplio rango de personas; en caso opuesto, deberíamos usar el algoritmo de detección de movimiento expuesto en el [apartado 4.3.2](#).

Consideremos en primer lugar el reconocimiento de esqueleto que realiza *Kinect* ([ilustración 25](#)):

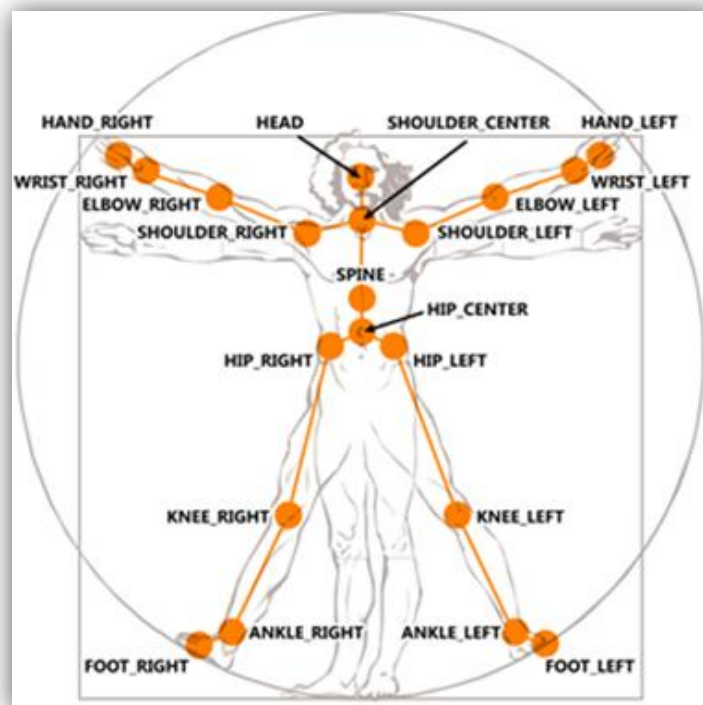


Ilustración 25: Articulaciones del esqueleto reconocidas por *Kinect*

Teniendo esto en cuenta y realizando algunos cálculos sobre los valores que devuelve *Kinect*, podemos obtener los llamados *joints* que serían las articulaciones del esqueleto, todos ellos con su altura, anchura y profundidad calculada. Una vez llevado esto a cabo, el algoritmo se basará en encontrar la diferencia entre los valores de los *joints* devueltos por *Kinect* y mirar cuál es la desviación necesaria para considerar que una de las partes del esqueleto que queremos reconocer ha cambiado su estado (por ejemplo, se puede inferir que comparando los *joints* pertenecientes a la cabeza y el tronco se obtendrá una buena estimación de si la cabeza está agachada, erguida o en su posición normal).

La forma general del algoritmo es la siguiente (ilustración 26):

```
Detectar_pose{  
  
    Calcular la desviación estimada en el eje X, y o Z para que se cumpla la nueva pose (desv_eje)  
  
    Restar la distancia entre las dos articulaciones implicadas en la pose (resta)  
  
    Si resta[eje] > desv_eje  
  
    ENTONCES Guardar_nueva_pose()  
  
}
```

Ilustración 26: Algoritmo de detección de pose corporal

4.3.2. Algoritmo de detección de movimiento

Como se mencionó en el apartado anterior, habrá ocasiones en las que la detección mediante la resta de *joints* será bastante complicada: cuando haya características no homogéneas que cambien mucho según el usuario observado, o desviaciones prácticamente imperceptibles mediante la información suministrada por el esqueleto (por ejemplo, en el caso de detectar que los hombros se encogen se obtiene por el algoritmo expuesto en la ilustración 26 una alta tasa de error).

Es entonces cuando se debe optar por esta alternativa, algo más compleja computacionalmente, que lo que hace es obtener varias imágenes del esqueleto con sus respectivos valores en altura, anchura y profundidad y comprobar la evolución de una articulación concreta en esta lista de imágenes, entendiéndose evolución como el cambio ocurrido de un *frame* (imagen en secuencia) a otro.

La forma general del algoritmo es la siguiente (ilustración 27):

```

Detectar_movimiento (Lista_posiciones_art) {

    Cargar el avance esperado de la articulación en el eje concreto por iteración (avance_art)

    Cargar el avance total esperado para que se considere el movimiento válido (avance_total)

    Cargar la duración mínima y máxima del movimiento (dur_min, dur_max)

    PARA (índice = 0; índice < Lista_posiciones_art; índice+1){

        SI (lista_posiciones_art[índice+1] - lista_posiciones_art[índice]) < avance_art

        ENTONCES Salir del bucle, ya que no ha avanzado en la dirección esperada

        SI NO {

            SI (lista_posiciones_art[índice] - lista_posiciones_art[0]) >= avance_total

            ENTONCES {

                Calcular_milisegundos_transcurridos(índice, 0)

                SI milisegundos_transcurridos > dur_min Y milisegundos_transcurridos < dur_max

                ENTONCES Guardar_nueva_pose()

            }

        }

    }

}

```

Ilustración 27: Algoritmo de detección de movimiento corporal

4.3.3. Algoritmo de detección facial

Para lograr la detección facial tanto de cejas como de boca se han usado los coeficientes de animación expresados en [17] y mencionados varias veces a lo largo del presente documento. Como recordatorio, los coeficientes o unidades de animación (AU por sus siglas en inglés) son desviaciones de la postura normal que adopta la cara en su reconocimiento. Queda mejor ilustrado con un ejemplo que también proviene de [17]: por ejemplo en este caso la AU 0 indica la elevación del labio superior (ilustración 28):

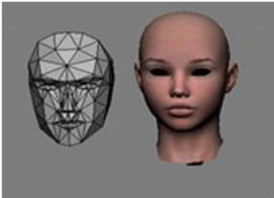

AU Name and Value	Avatar Illustration	AU Value Interpretation
Neutral Face (all AUs 0)		
AU0 – Upper Lip Raiser (In Candid3 this is AU10)		0=neutral, covering teeth 1=showing teeth fully -1=maximal possible pushed down lip

Ilustración 28: Ejemplo de unidad de animación

Para la detección de los gestos faciales, por tanto, el algoritmo usado ha sido una secuencia de sentencias condicionales y se han llevado a cabo varias pruebas con valores experimentales de cada unidad de animación (que oscilan entre -1 y 1) hasta que los resultados han sido suficientemente precisos como para considerarlo una implementación válida.

4.3.4. Algoritmo de inferencia de emociones

En cuanto a la detección de emociones, se ha desarrollado un algoritmo basado en el patrón que aparece en la tabla que se planteó en el análisis (Apartado 3.3) para la implementación de la clase *MotorInferencias*; a saber:

	Aversión	Sorpresa	Alegría	Tristeza
Cabeza	Erguida	Al frente	Erguida /al frente	Agachada
Brazos	-	Arriba / tapando cara	Arriba	Tapando cara / atrás / sueltos
Espalda	Atrás	-	Normal / Atrás	Adelantada
Hombros	Atrás / encogidos	Atrás / encogidos	-	-
Boca	-	Abierta	Sonrisa	Cerrada
Cejas	Abajo	Arriba	Arriba	-

Tabla 30: Algoritmo de clasificación de gestos en emociones

4.4. Interfaz de usuario utilizada

La interfaz de usuario será el medio con el cual el usuario pueda interaccionar con la aplicación. Cabe señalar que estará basada en el análisis realizado en el [apartado 3.6](#) (definición de interfaces de usuario) del presente documento, mediante el cual se ha escogido la tercera alternativa expuesta en ese punto y se ha pasado de un boceto a la implementación real.

El resultado de una ejecución sería el siguiente:



Ilustración 29: Interfaz de usuario final

A continuación, se detallarán las zonas numeradas en la ilustración 29:

- 1) Zona de la cámara: coincide con el requisito de software **RS-01**, y se trata de ofrecer al usuario una imagen de su propia grabación para seguir correctamente el proceso y que se tenga respuesta del sistema.
- 2) Zona de las emociones: muestra un porcentaje de cada una de las emociones básicas consideradas para este proyecto (aversión, sorpresa, alegría y tristeza) incluyendo porcentajes y barras de progreso (como se expuso en el requisito **RS-02**) para la emoción percibida en la iteración actual. Como las iteraciones se realizan en una franja corta de tiempo, se incluye además un campo denominado “emoción global” que almacena la emoción más frecuente ocurrida cada dos segundos, como manifestaba el requisito **RS-08**.
- 3) Fiabilidad de medición: satisface lo expuesto en el requisito **RS-07**: es un indicador que cambiará conforme esté el usuario situado frente al Kinect; el icono que lo acompaña mostrará una cara feliz, indiferente o triste en función de la idoneidad de la distancia en que se encuentre el usuario.
- 4) Tracking del cuerpo: se corresponde con lo indicado en el requisito **RS-03**, mostrando el estado de cada una de las partes que se han considerado implicadas en el cambio emocional; a saber: cabeza, hombros, espalda, brazos, boca y cejas, de manera que el usuario pueda precisar la exactitud de esta información.

Por último, cabe señalar que con esta distribución espacial de la interfaz se da por satisfecho el requisito **RS-12**, que habla sobre la necesidad de tener una interfaz intuitiva, considerándose que en este caso es capaz de ser interpretada de un primer vistazo.

5. Resultados, trabajo futuro y conclusiones

Este punto sirve como cierre del proyecto, después de haber explorado varios trabajos relacionados con él y las tecnologías involucradas en el proyecto (estado del arte), haber llevado a cabo una definición de requisitos y una definición del sistema en el análisis, haber considerado su modelo detallado de clases y su interfaz completa en la parte del diseño y haber llevado a cabo la implementación del sistema.

5.1. Resultados

En este apartado se revisarán los objetivos conseguidos que figuran en el [apartado 1.2](#) del presente documento.

En primer lugar se revisará el éxito logrado en los objetivos secundarios:

- Conseguir hacer detección precisa del cuerpo: se ha conseguido establecer un análisis del cuerpo bastante robusto y probado al analizar la cabeza, la espalda, los hombros y los brazos, pudiendo aparecer algún falso positivo (de manera ocasional y normalmente debido a una mala posición del usuario frente a la *Kinect*) en la detección de brazos. Este objetivo se considera plenamente satisfecho siempre y cuando el usuario esté frente a la cámara y en un rango de distancia considerada fiable (descrito por el parámetro “fiabilidad de medición” que figura en la interfaz, cuya lógica se puede consultar en el [apartado 4.4](#)).
- Conseguir hacer detección facial: el análisis facial ha sido algo más complejo debido a que la distancia considerada razonable para una medición correcta era de 1,5 metros respecto del *Kinect*; aun así, se puede contrastar su buen funcionamiento mediante la ejecución de la aplicación implementada en este proyecto. De modo general, existe una tasa alta de acierto en los estados de las cejas (arriba, centro y abajo) y en la detección de si la boca está abierta o cerrada, siendo un poco menor dicha tasa en la comprobación de si el usuario está sonriendo. Este objetivo se considerará satisfecho al darse unas condiciones análogas al análisis corporal.
- Almacenar la información extraída en un soporte secundario: en este caso se almacenará única y exclusivamente la información sobre el movimiento del análisis facial y corporal almacenándolos cada segundo; se ha resuelto no guardar en este caso las emociones inferidas ya que esta ponderación sería subjetiva en función del experto que consultase estos ficheros, así que se ha decidido no condicionar esto para crear unos datos veraces en cada interacción con el usuario.

- Inferir emociones a partir de la detección gestual: se considera que, tras haber tenido en cuenta los documentos del apartado 3.3 sobre el movimiento corporal relacionado con las emociones [33] [34], y haber seguido una correcta interpretación del modelo *Candide3* sobre el reconocimiento facial y sus consideraciones en el entorno de desarrollo de *Kinect* [16] [35], se han realizado con éxito algunas consideraciones comprobadas empíricamente sobre la importancia en cada emoción de una pose o gesto concreto; esto ha sido transformado en un porcentaje que cambia cada iteración y en una emoción global que se mantiene durante varias iteraciones para dar una perspectiva más amplia y correcta sobre las emociones del usuario. Cabe señalar que los porcentajes de las emociones están basados en los gestos realizados por el usuario, por lo que el porcentaje de cada una de ellas es **independiente del porcentaje de las otras** y, por último, que la detección de alegría llega pocas veces al 100%, ya que al ser manifestada por el usuario la detección facial presenta imprecisiones.

Ahora, se procede a analizar el cumplimiento del objetivo principal que es conseguir una **inferencia de emociones a partir de patrones definidos y de manera contrastable (almacenando dicha información)**: en principio este objetivo estaría logrado ya que subjetivamente da una buena estimación con varias personas de distintos rasgos físicos (probados con el autor del proyecto y dos personas más), además de realizar el mencionado almacenamiento de información. El paso que quedaría para corroborar que se ha logrado dicho objetivo sería **contrastar la validez de información** mediante expertos fiables que pudiesen analizar la corrección de las inferencias realizadas para detectar las emociones y aportar ideas, y una larga batería de pruebas con usuarios finales que pudiesen hacer el programa lo más genérico posible.

Se exponen de todos modos como aproximación fiable al objetivo principal: un fragmento de la información almacenada en XML sobre los movimientos corporales (ilustración 30) y resultados de las inferencias llevadas a cabo sobre las cuatro emociones: aversión, sorpresa, alegría y tristeza (ilustraciones 31, 32, 33 y 34, respectivamente):

```
- <pose>
  <fecha>11/08/2012 17:32:29</fecha>
  <cabeza>al frente</cabeza>
  <espalda>al frente</espalda>
  <hombros>al frente</hombros>
  <brazos>sueltos</brazos>
  - <cara>
    <boca>cerrada</boca>
    <cejas>centro</cejas>
  </cara>
</pose>
- <pose>
  <fecha>11/08/2012 17:32:30</fecha>
  <cabeza>al frente</cabeza>
  <espalda>hacia delante</espalda>
  <hombros>hacia atrás</hombros>
  <brazos>sueltos</brazos>
  - <cara>
    <boca>cerrada</boca>
    <cejas>abajo</cejas>
  </cara>
</pose>
```

Ilustración 30: Ejemplo del XML generado por la aplicación



Ilustración 31: Detección de aversión

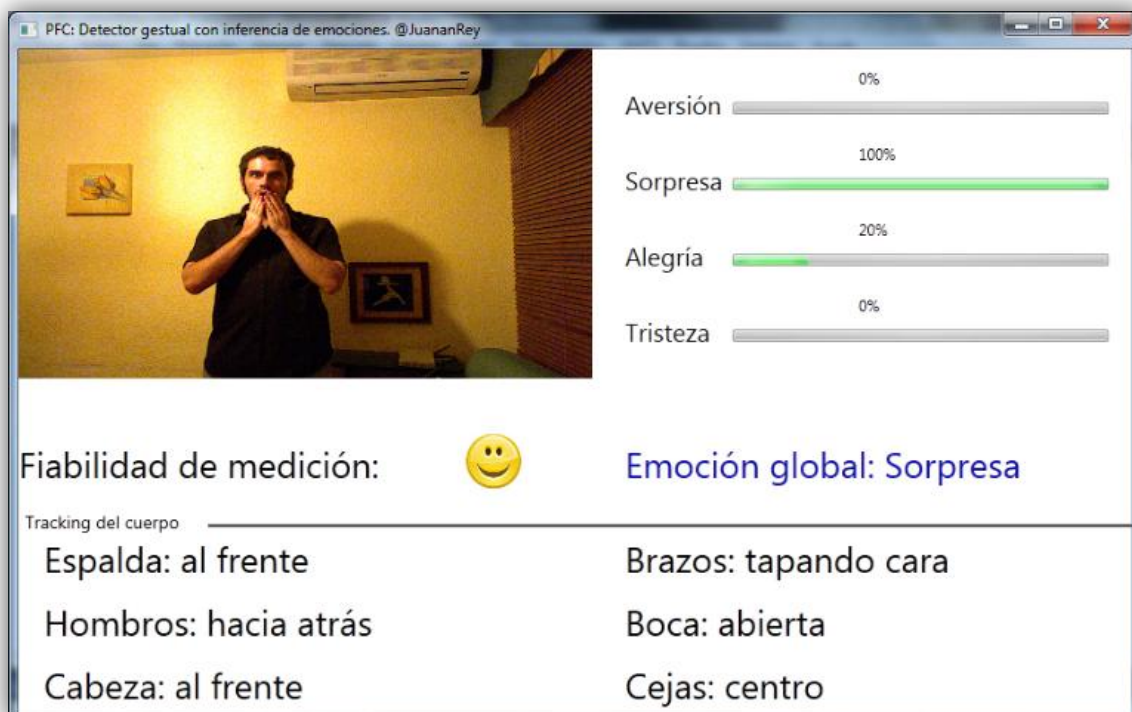


Ilustración 32: Detección de sorpresa



Ilustración 33: Detección de alegría

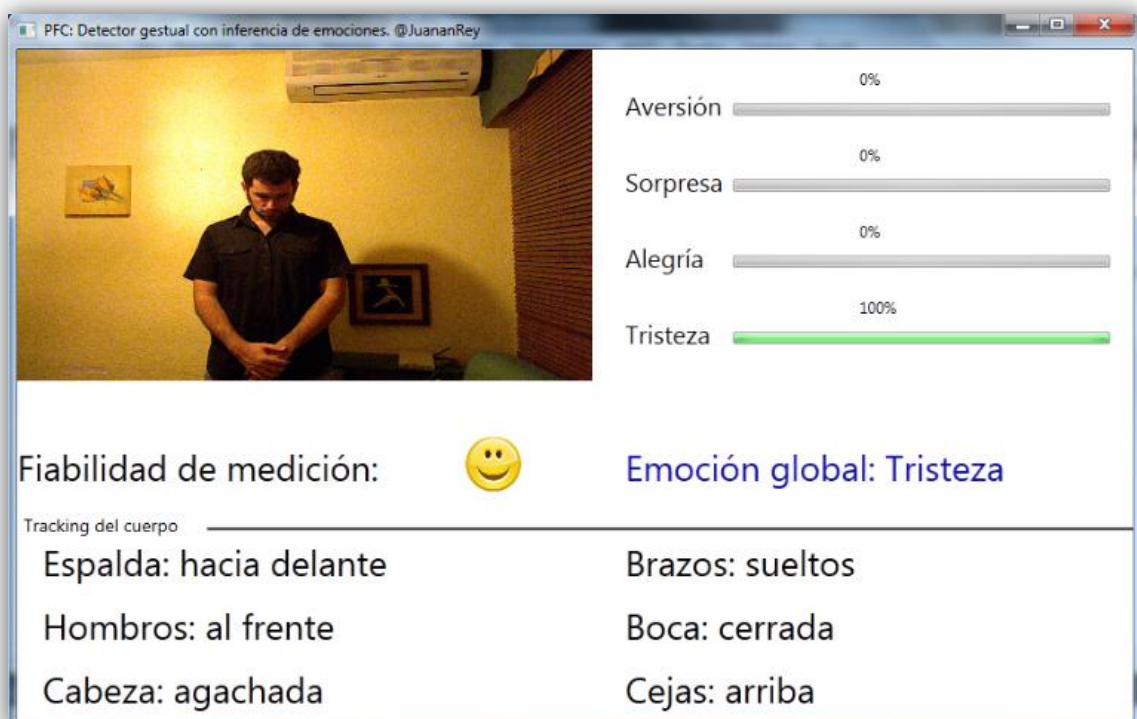


Ilustración 34: Detección de tristeza

5.2. Trabajo futuro

En el caso de que alguien interesado en el proyecto quisiera retomar y continuar con él, los desarrollos futuros que se podrían llevar a cabo con el mismo serían los siguientes:

- Almacenamiento en soporte secundario de las emociones inferidas cada segundo.
- Análisis exhaustivos y contrastados con personal competente del área de la psicología para la asignación de pesos en el motor de inferencias, buscando una detección más precisa.
- Posibles mejoras y correcciones en la detección facial (complicada en el momento de realizar este proyecto cuando la distancia entre la cámara y el usuario, para su correcto funcionamiento, debe ser mayor o igual que 1,50 m).
- Unión con otro detector de emociones que utilice el canal auditivo para aumentar la precisión.
- Detección de emociones de varios usuarios simultáneamente para poder inferir cómo se desarrolla el proceso de comunicación.
- Considerar la velocidad a la que se realizan los movimientos para clasificar con más precisión las emociones (como se mencionaba en [8], movimientos más rápidos suelen asociarse con alegría o enfado y movimientos más lentos con un estado triste).

5.3. Conclusiones

Llegados a este último punto, quiero dejar a título personal algunas conclusiones extraídas después de haber trabajado en este proyecto; en primer lugar hablaré sobre las tecnologías y sus aportes, y en segundo lugar daré mi visión personal sobre el desarrollo del proyecto.

Al empezar a probar el *Kinect* con el primer entorno de desarrollo de Microsoft para *Kinect* (versión 1.0), me llevé una grata sorpresa; en general, he considerado a lo largo del proyecto que he trabajado con una tecnología potente y sencilla: la de Microsoft, cuyo entorno de desarrollo *Visual Studio* y cuyo lenguaje de programación *C#* he tenido que aprender prácticamente desde el principio, y que gracias a los conocimientos previos obtenidos en *Java* y otros lenguajes (unos variopintos y útiles, otros no tanto) durante la carrera han ocasionado que la dificultad de aprendizaje de *C#* fuera muchísimo menor de lo esperado.

La parte negativa que podría señalar de las tecnologías con las que he trabajado es que al pasar el entorno de desarrollo oficial de *Kinect* de la versión 1.0 a la 1.5 no fuese compatible hacia atrás, lo cual debería ser revisado y cuya actualización de modo no traumático está expuesto en mi blog sobre *Kinect* “Luchando con *Kinect*” [38], creado a propósito de este proyecto.

He tenido que poner en práctica diversos conocimientos sobre ingeniería del software a la hora de documentar el proyecto, cuyo aprendizaje me ha sido muy útil a la hora de poder crear nuevo material. En el estado del arte he tenido que revisar mis matemáticas para entender cómo funciona el reconocimiento sonoro y las señales entre otras cosas, y los (por no haberlos contado) infinitos algoritmos desarrollados en estos seis años han servido en gran parte de base para emprender este proyecto sin perderme.

Creo que he conseguido perfeccionar la elaboración de esquemas explicativos que son capaces de sintetizar mucha información en una sola imagen, ya que el proyecto está bastante marcado por ellos, y he aprendido mucho sobre áreas y tecnologías que me eran completamente ajenas y que tienen un uso actualmente, como puedan ser las tecnologías *.Net*, el sensor *Kinect* y el área de *Affective Computing*, lo cual ha tenido en mis inquietudes un impacto muy positivo.

No todo fue tan positivo: en algunos momentos no conseguí ver el final del proyecto, ya que en un principio pensaba dejarlo solamente como detector corporal ya que me resultó una combinación perfecta entre la originalidad de hacer algo nuevo (no encontré ningún detector corporal para inferir emociones) y las capacidades de *Kinect*, reparando algo más tarde en que esta detección era insuficiente para dicha inferencia, hasta que apareció la versión 1.5 del entorno de desarrollo *Kinect* que mencioné, y que incluía reconocimiento de los rasgos faciales, de modo que pude retomar el proyecto con una alta tasa de acierto en las emociones detectadas.

Espero muy sinceramente que mi pequeña aportación pueda servir a alguien que trabaje en esta área de *Affective computing*, que está en un momento muy interesante como se expuso en el estado del arte.

A título personal, quería señalar que nunca pensé que la idea que tuve con los cascos puestos volviendo de la universidad en el metro, cuando vi a dos personas discutiendo y reconciliándose en cuestión de segundos y pensé por primera vez en la detección de emociones, pudiese llevarse a cabo con la tecnología actual. Así pues, por casualidad hablé con José Manuel (compañero de la universidad) de ello mientras me tomaba un café y me ofreció prestarme la *Kinect*, cuyas capacidades tampoco conocía hasta el momento. En un primer momento no tenía claro el hecho de realizar la detección con el cuerpo ya que apenas había documentación y hablé con mi tutor que había ofertado un proyecto sobre emociones, hasta que él encontró un documento que relacionaba la expresión de emociones en avatares en gestos: en ese momento exacto empecé a tener verdaderas esperanzas en ello.

Y así fue como, tras una serie de coincidencias, pude terminar trabajando en este proyecto: con toda la ilusión que supone no saber muy bien hacia dónde se camina, pero viendo que las barreras van quedando atrás.

Bibliografía

- [1] *Habilidades de comunicación: técnicas para la comunicación eficaz*. Disponible [Internet]: http://www.psicologia-online.com/monografias/5/comunicacion_eficaz.shtml [10-06-2012]
- [2]: Picard, R.W. (1997): *Affective Computing*. MIT Press.
- [3]: Picard, R.W. (1999): *Affective Computing for HCI*. Disponible [Internet]: <http://affect.media.mit.edu/pdfs/99.picard-hci.pdf> [17-06-2012]
- [4]: *Programación y reprogramación neurolingüística*. Disponible [Internet]: http://www.degerencia.com/articulo/dpnl_mas_rpnl_desprogramacion_y_reprogramacion_neurolinguistica [26-09-2012]
- [5]: Haggbloom, S. J. et al. (2002): *The 100 Most Eminent Psychologists of the 20th Century*. Disponible [Internet]: <http://www.assessmentpsychology.com/eminentspsychologists.htm> [26-09-2012]
- [6]: Gunes, H., Piccardi, M., Pantic, M. (2008): *From the Lab to the Real World: Affect Recognition Using Multiple Cues and Modalities*. Disponible [Internet]: http://doc.utwente.nl/65266/1/pantic_from_the_lab_book_chapter.pdf [17-06-2012]
- [7]: Hoque, M. E., Morency, L-P, Picard, R. W. (2011): *Are you friendly or just polite? - Analysis of smiles in spontaneous face-to-face interactions*. Disponible [Internet]: <http://affect.media.mit.edu/pdfs/11.Hoque-Morency-Picard-Polite-Amused-Smiles-ACII.pdf> [17-06-2012]
- [8]: Dittrich, W.H., Atkinson, P.H. (2008): *The Perception of Bodily Expressions of Emotion and the Implications for Computing*. Disponible [Internet]: http://www.intechopen.com/books/affective_computing/the_perception_of_bodily_expressions_of_emotion_and_the_implications_for_computing [26-09-2012]
- [9]: *Affective Computing website*. Disponible [Internet]: <http://affect.media.mit.edu/publications.php>
- [10]: Ambady, N., Rosenthal, R. (1992): *Thin Slices of Expressive Behavior as Predictors of Interpersonal Consequences: A Meta-Analysis*. Disponible [Internet]: <http://ambadylab.stanford.edu/pubs/1992Ambady.pdf> [19-06-2012]
- [11]: Camurri, A. et al. (2004): *Expressive interfaces*.
- [12]: Bänziger, T., Scherer, K. (2007): *Using actor portrayals to systematically study multimodal emotion expression: the gemep corpus*, páginas 476-487.
- [13]: Elgammal, A.; Shet, V.; Yacoob, Y., Davis, L.S. (2003): *Learning dynamics for exemplar based gesture recognition*. Disponible [Internet]: <http://www.cfar.umd.edu/~yaser/cvpr03.pdf> [26-09-2012]

- [14]: Höök, Kristina (2012): *Affective Computing: Affective Computing, Affective Interaction and Technology as Experience*. Disponible [Internet]:
http://www.interaction-design.org/encyclopedia/affective_computing.html [26-09-2012]
- [15]: McDuff, D. (2011): *Acume: A Novel Visualization Tool for Understanding Facial Expression and Gesture Data*. Disponible [Internet]:
<http://affect.media.mit.edu/pdfs/11.McDuff-et-al-Acume-2011.pdf> [22-06-2012]
- [16]: *Candide3: Modelo de construcción facial*. Disponible [Internet]:
<http://www.icg.isy.liu.se/candide/> [26-09-2012]
- [17]: *Documentación MSDN sobre la funcionalidad de Face Tracking*. Disponible [Internet]:
<http://msdn.microsoft.com/en-us/library/jj130970.aspx> [10-09-2012]
- [18]: D' Mello, S. et al. (2008): *AutoTutor Detects and Responds to Learners Affective and Cognitive States*. Disponible [Internet]:
<http://affect.media.mit.edu/pdfs/08.dmello-et-al-autotutor.pdf> [30-06-2012]
- [19]: *Proyecto FaceSense: análisis facial extendido*. Disponible [Internet]:
<http://trac.media.mit.edu/mindreader/> [30-06-2012]
- [20]: Conde, P. et al. (2000): *Estabilización de la actividad electrodermal y ciertos parámetros respiratorios*. Disponible [Internet]:
<http://www.psicothema.com/psicothema.asp?id=352> [05-07-2012]
- [21]: Ayzenberg, Y., Hernández, J., Picard, R.W. (2012): *FEEL: Frequent EDA and Event Logging – A Mobile Social Interaction Stress Monitoring System*. Disponible [Internet]:
<http://affect.media.mit.edu/pdfs/12.Ayzenberg-Hernandez-Picard-CHI.pdf> [05-07-2012]
- [22]: Autor no referenciado, 2011: *PFC - Expresión de emociones*. *Speech Technology Group, UPM*. Disponible [Internet]:
<http://lorien.die.upm.es/juancho/pfcs/GMS/cap2.pdf> [05-07-2012]
- [23]: Resa, C.O. (2009): *PFC- Detección de emociones en voz espontánea, UAM*. Disponible [Internet]: http://atvs.ii.uam.es/files/2009_0722CarlosOrtego.pdf [05-07-2012]
- [24]: Munoz, S.H et al. (2011): *RFuzzy: Syntax, semantics and implementation details of a simple and expressive fuzzy tool over Prolog*.
- [25]: *Argentina investiga: un software identifica las emociones del hablante*. Disponible [Internet]:
http://infouniversidades.siu.edu.ar/noticia.php?titulo=un_software_identifica_las_emociones_del_hablante&id=235 [05-07-2012]
- [26]: *Pulso social: Emospeech, el software de reconocimiento de las emociones de la voz*. Disponible [Internet]:
<http://pulsosocial.com/2012/07/18/emospeech-el-software-de-reconocimiento-de-las-emociones-de-la-voz/> [05-07-2012]

- [27]: Poh, M., McDuff, D., Picard, R.W. (2010): *Non-contact, automated cardiac pulse measurements using video imaging and blind source separation*. Disponible [Internet]: <http://www.opticsinfobase.org/oe/abstract.cfm?uri=oe-18-10-10762> [08-07-2012]
- [28]: *Información del sensor 3D PrimeSense*. Disponible [Internet]: <http://www.primesense.com/press-room/resources/file/4-primesense-3d-sensor-data-sheet> [08-07-2012]
- [29]: *Recursos para el Hand Tracking en Kinect*. Disponible [Internet]: <http://42spikes.com/post/Kinect-Hand-Tracking-Resources.aspx> [08-07-2012]
- [30]: *OpenNI website*. Disponible [Internet]: <http://openni.org/> [09-07-2012]
- [31]: *Channel9 website*. Disponible [Internet]: <http://channel9.msdn.com/> [26-09-2012]
- [32]: *Métrica V3 website*. Disponible [Internet]: http://administracionelectronica.gob.es/?_nfpb=true&_pageLabel=P60085901274201580632&langPae=es [11-08-2012]
- [33]: A. Guedes et al. (2008): *Animación de Actores Virtuales: Expresión de Emociones Mediante Lenguaje Corporal*. Disponible [Internet]: <http://giga.cps.unizar.es/affectivelab/publications/CEIG08Abner.pdf> [13-08-2012]
- [34]: Su et al. (2007): *Personality and Emotion-based High-level Control of Affective Story Characters*. Disponible [Internet]: <http://eprints.qut.edu.au/3818/1/3818.pdf> [13-08-2012]
- [35]: Descarga del entorno *Kinect for Windows SDK 1.5*. Disponible [Internet]: <http://go.microsoft.com/fwlink/?LinkID=246662> [22-09-2012]
- [36]: *Micromovimiento: Reconocer el miedo y la sorpresa en otros*. Disponible [Internet]: <http://micromovimiento.com/?p=2274> [17-08-2012]
- [37]: *UML website*. Disponible [Internet]: <http://www.uml.org/> [26-09-2012]
- [38]: *Luchando con Kinect website*. Disponible [Internet]: <http://luchandoconkinect.blogspot.com.es/> [26-09-2012]

Anexo A: Glosario de términos.

AU: siglas en inglés de “Unidad de animación”, medida que analiza las variaciones faciales en función de su estado neutral.

BPMS: siglas en inglés de “Sistema de medida de la presión corporal”.

Candide3: estándar para el modelo facial que construye una cara a partir de un número no muy grande de polígonos.

CMD-xx: acrónimo de Clase del Modelo Detallado, que aparece de la siguiente forma: CMD-Número.

CU-xx: acrónimo de Casos de Uso, que aparece de la siguiente forma: CU-Número.

C#: lenguaje de programación orientada a objetos.

Data stream: flujo de datos.

DBN: siglas en inglés de “Red bayesiana dinámica”, red bayesiana cuyas inferencias se propagan a través del tiempo.

EDA: siglas en inglés de “Actividad electrodermal”, medida sobre el nivel de activación del sujeto.

Feedback: realimentación, reacción o respuesta de un receptor respecto a la actuación del emisor.

Frames: imagen particular dentro de una sucesión de imágenes.

HCI: siglas en inglés de “Interacción persona-ordenador”, referidas al intercambio que se produce entre ambos.

HMM: siglas en inglés de “Modelo oculto de Markov”, modelo estadístico que determina el valor de algunos parámetros desconocidos (ocultos) a partir de otros observables.

IDE: siglas en inglés de “Entorno integrado de desarrollo”, se trata de un *SDK* con una interfaz de usuario, editor de texto y otras funcionalidades extra.

Joint: articulación. En este caso se refiere a las articulaciones detectadas por el *Kinect* en el *tracking* del esqueleto.

Kinect: sensor hardware que puede actuar como cámara tridimensional y como micrófono.

Métrica V3: metodología de sistemas de información que se encarga de su planificación, mantenimiento y desarrollo.

OpenNI: entorno de desarrollo para escribir aplicaciones mediante interacción natural.

Paper: texto científico, generalmente breve, para su publicación en revistas especializadas.

Pitch: frecuencia a la que oscilan nuestras cuerdas vocales al omitir sonidos en un estado no alterado.

Píxel: unidad mínima de color en forma de punto que aparece en una imagen de forma homogénea.

PYMES: acrónimo de “Pequeñas Y Medianas Empresas”.

RAM: siglas en inglés de “Memoria de acceso aleatorio”, memoria temporal que utilizan los ordenadores para acceder con cierta rapidez a los datos.

RGB: modelo de color en el que se indica la cantidad (en un rango de 0 a 255) de rojo, verde y azul de un pixel de la imagen.

RS-xx: acrónimo de Requisito de Software, que aparece de la siguiente forma: RS-Número.

SDK: siglas en inglés de “Kit de desarrollo software”, conjunto de herramientas destinadas a desarrollar software.

Tracking: seguimiento o rastreo.

UI: siglas en inglés de “Interfaz de Usuario”, medio del usuario para interactuar con el sistema.

Visual Studio: IDE de Microsoft.

XML: siglas en inglés de “Lenguaje extendido de marcado”, lenguaje que permite definir la gramática de otros y con soporte a bases de datos.

.NET: entorno de trabajo de Microsoft.

Anexo B: Planificación

En este punto se va a establecer una comparativa entre la planificación hecha *a priori* en el documento y la planificación final, ambas mostradas en un Diagrama de Gantt. Estas planificaciones se han realizado en función a las consideraciones hechas en el Apartado 1.4. (Pasos realizados en el proyecto) del presente documento. Como recordatorio, se ha decidido dividir el proyecto en tres grandes fases:

- Estudio del problema.
- Documentación.
- Implementación.

Dado que estas fases no se han desarrollado de manera secuencial, sino que se ha vuelto atrás al descubrir nuevas necesidades se va a dividir el proyecto en las siete actividades mencionadas en el Apartado 1.4 en lugar de en las tres grandes frases ya que es mucho más sencillo establecer una división temporal; a saber:

- 1) Estudio sobre la parte no técnica del problema.
- 2) Estudio sobre la parte técnica del problema.
- 3) Comienzo de documentación.
- 4) Implementación de la primera versión.
- 5) Implementación de la segunda versión.
- 6) Final de documentación (conclusiones y anexos) y revisión de la implementación.
- 7) Fases de pruebas finales de implementación y revisión de la documentación.

A continuación se mostrará la tabla y el diagrama de Gantt de la planificación estimada:

Actividades	Fecha de inicio	Duración (días)	Fecha de fin
Actividad 1	09/02/2012	21	29/02/2012
Actividad 2	01/03/2012	14	14/03/2012
Actividad 3	15/03/2012	56	09/05/2012
Actividad 4	10/05/2012	49	27/06/2012
Actividad 5	28/06/2012	35	01/08/2012
Actividad 6	02/08/2012	14	15/08/2012
Actividad 7	16/08/2012	7	22/08/2012

Tabla 31: Planificación estimada del proyecto

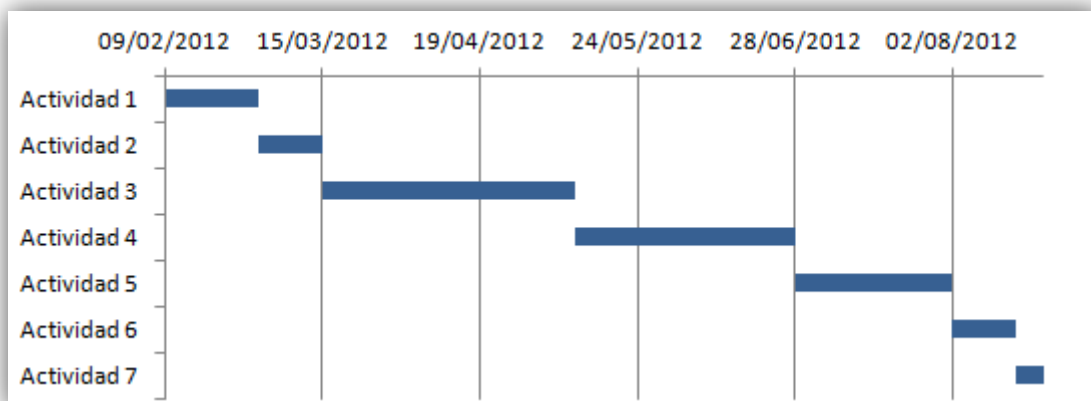


Ilustración 35: Diagrama de Gantt de la planificación estimada del proyecto

Y este sería el diagrama de la planificación real (como se puede ver, hay algunos cambios, entre otros un mes y cinco días de retraso debido a las vacaciones de verano que coincidieron con la etapa de implementación y pequeñas paradas, pero en general guarda bastante similitud):

Actividades	Fecha de inicio	Duración (días)	Fecha de fin
Actividad 1	09/02/2012	24	03/03/2012
Actividad 2	04/03/2012	15	18/03/2012
Actividad 3	19/03/2012	43	30/04/2012
Actividad 4	01/05/2012	54	23/06/2012
Actividad 5	24/06/2012	65	27/08/2012
Actividad 6	28/08/2012	18	14/09/2012
Actividad 7	15/09/2012	13	27/09/2012

Tabla 32: Duración real del proyecto

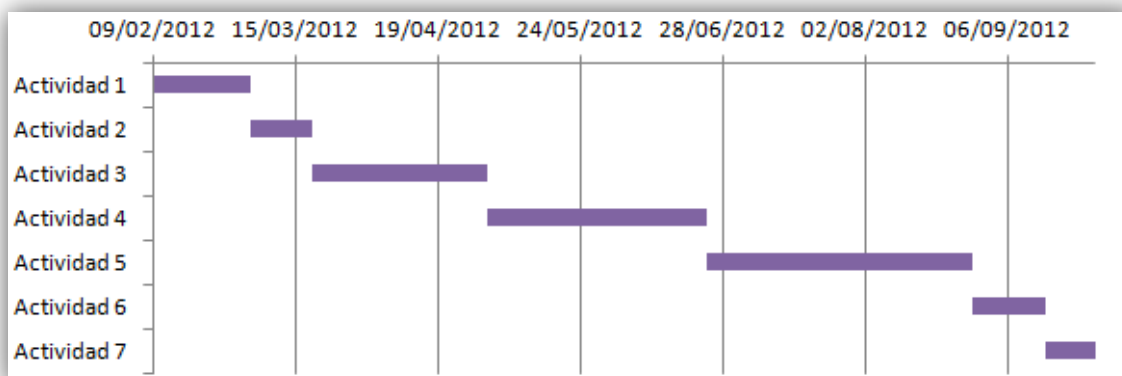


Ilustración 36: Diagrama de Gantt con la duración real del proyecto

Anexo C: Presupuesto

En este último anexo se van a presentar los costes relacionados con el proyecto, introduciendo finalmente una tabla de resumen propuesta como modelo por la Universidad Carlos III de Madrid. Todos los costes calculados incluyen el IVA.

Gastos de personal

Aquí figuran los costes de personal. El autor de este proyecto, Juan Antonio Rey Mendo, ha asumido los roles de: jefe de proyecto, responsable de documentación, analista, diseñador y programador de la aplicación. Se asumirá que las funciones de analista las haré como ingeniero senior y las funciones de diseño como ingeniero, y que el trabajo ha sido (una aproximación bastante realista) de 2 horas por día, durante 7,67 meses (como se expuso en el [Anexo C: Planificación](#)).

El gasto total figura en la siguiente tabla:

Categoría	Coste/hora (€)	Horas totales (estimadas)	Dedicación (hombre / mes)*	Coste total (€)
Responsable de documentación	15	100	1,666666667	1500
Ingeniero Senior	30	40	0,666666667	1200
Ingeniero	20	70	1,166666667	1400
Programador	10	250	4,166666667	2500
TOTAL		450	7,67	6600

Tabla 33: Gastos de personal

* Un hombre / mes, considerando que se han trabajado dos horas al día, trabaja 60 horas

Gastos de hardware

El hardware empleado para esta aplicación figura en el [Apartado 3.2.2. Identificación del entorno tecnológico](#) del presente documento. A continuación se muestra la tabla con estos gastos:

Hardware	Coste/ unidad(€)	Unidades	Coste total(€)	Coste Imputable(€)
Ordenador Fujitsu Amilo Pi 3540	599	1	599	13,75
Sensor Kinect para Xbox 360.	110	1	110	74,88
TOTAL				88,63

Tabla 34: Gastos de hardware

Siendo la amortización: $\frac{A*C*D}{B}$, donde:

- A = nº de meses desde la fecha de facturación.
- B = periodo de depreciación (60 meses).
- C = coste del equipo.
- D = % de uso que se dedica al proyecto.

Gastos de software y licencias

El software empleado para esta aplicación figura en el Apartado 3.2.2. Identificación del entorno tecnológico del presente documento. A continuación se muestra la tabla con estos gastos (en este caso no se ha considerado la amortización):

Software	Coste/ unidad(€)	Unidades	Coste total(€)
Visual Studio Professional 2010	1558	1	1558
Office 2010	90	1	90
Altova Umodel 2012 Enterprise	299	1	299
TOTAL			1947

Tabla 35: Gastos de software

Gastos de material fungible

Se entiende por material fungible todo aquel que se gasta y debe ser repuesto, por ejemplo todo aquel material de oficina: folios, recambios de impresora. El gasto asociado ha sido de **100 Euros** aproximadamente, teniendo en cuenta **conexión a internet** mensual a un muy bajo precio.

Resumen de gastos totales

A continuación se expondrán los gastos totales asociados al proyecto, teniendo en cuenta los cálculos hechos a lo largo de este anexo:

Concepto	Coste (€)
Personal	6600
Hardware	88,63
Software	1947
Material fungible	100
TOTAL (sin gastos indirectos)	8735,63
Gastos indirectos (+20% total)	1747,126
TOTAL	10482,756

Tabla 36: Gastos totales asociados al proyecto

Plantilla de resumen

Además de esta plantilla, se incluye en el CD entregado en el proyecto este presupuesto a una resolución mayor para hacerlo legible, documento con la signature "Presupuesto.pdf".



UNIVERSIDAD CARLOS III DE MADRID
Escuela Politécnica Superior

PRESUPUESTO DE PROYECTO

1-Autor:

Rey Mendo, Juan Antonio

2-Departamento:

Informática

3-Descripción del Proyecto:

- Título: INFERENCIA DE EMOCIONES A TRAVÉS DE DETECCIÓN CORPORAL Y FACIAL
- Duración (meses): 7,67
- Tasa de costes indirectos: 28%

4- Presupuesto total del Proyecto (valores en Euros):

10482,756 Euros

5- Desglose presupuestario (costes directos)

PERSONAL

Apellidos y nombre	N.I.F. (no rellenar - solo a título informativo)	Categoría	Dedicación € (hombres mes)	Coste hombre mes	Coste (Euros)	Firma de conformidad
Rey Mendo, Juan Antonio		Responsable doc.	1,00000000	900,00	1.500,00	
Rey Mendo, Juan Antonio		Ingeniero senior	0,00000000	1.800,00	1.200,00	
Rey Mendo, Juan Antonio		Ingeniero	1,10000000	1.200,00	1.400,00	
Rey Mendo, Juan Antonio		Programador	4,10000000	600,00	2.500,00	
Hombres mes 7,06666662				Total	6.600,00	

⁴¹ 1 Hombre mes = 60 horas. Máximo anual de dedicación de 12 hombres mes (1575 horas)
Máximo anual para PDI de la Universidad Carlos III de Madrid de 8,8 hombres mes (1.155 horas)

EQUIPOS

Descripción	Coste (Euros)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste Imputable ⁴²
Kinect para Xbox 360	110,00	100	7,50	60	13,75
Ordenador Fujitsu Amilo P1 3540	599,00	100	7,50	60	74,88
		100		60	0,00
		100		60	0,00
		100		60	0,00
Total					88,63

⁴² Fórmula de cálculo de la Amortización:

$$\frac{A}{B} \times C \times D$$

A = nº de meses desde la fecha de facturación en que el equipo es utilizado
B = periodo de depreciación (60 meses)
C = coste del equipo (sin IVA)
D = % del uso que se dedica al proyecto (habitualmente 100%)

SUBCONTRATACIÓN DE TAREAS

Descripción	Empresa	Coste Imputable
No procede		
Total		0,00

OTROS COSTES DIRECTOS DEL PROYECTO⁴³

Descripción	Empresa	Costes Imputable
Visual Studio Professional 2010	Microsoft	1.558,00
Office 2010	Microsoft	90,00
Altova Umodel Enterprise 2012	Altova	299,00
Materiales fungibles	No procede	100,00
Total		2.047,00

⁴³ Este capítulo de gastos incluye todos los gastos no contemplados en los conceptos anteriores, por ejemplo: fungibles, viajes y dietas, otros,...

6- Resumen de costes

Presupuesto Costes Totales	Presupuesto Costes Totales
Personal	6.600
Amortización	89
Subcontratación de tareas	0
Costes de funcionamiento	2.047
Costes indirectos	1.747
Total	10.483